

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra Elektroniky

Aplikace umělých neuronových sítí v řízení elektrických
pohonů
Applications of Artificial Neural Networks in Control of Electrical
Drives

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra elektroniky

Zadání diplomové práce

Student: **Bc. Jakub Ščamba**
Studijní program: N2649 Elektrotechnika
Studijní obor: 2612T015 Elektronika
Téma: Aplikace umělých neuronových sítí v řízení elektrických pohonů
Applications of Artificial Neural Networks in Control of Electrical Drives

Zásady pro vypracování:

1. Teoretický rozbor aplikací umělých neuronových sítí v řízení elektrických pohonů.
2. Výběr vhodných metod využívajících umělé neuronové sítě pro řízení nebo estimaci veličin elektrických pohonů.
3. Návrh a realizace simulačních modelů vybraných částí regulační struktury pomocí programu Matlab-Simulink.
4. Simulační ověření modelů.
5. Analýza výsledků simulací a jejich porovnání s teoretickými předpoklady.

Seznam doporučené odborné literatury:

- [1] Brandštetter P.: Střídavé regulační pohony - moderní způsoby řízení, VŠB-TU Ostrava, 1999, ISBN 80-7078-668-X.
- [2] Brandštetter P. a kol.: Research of Artificial Intelligence Applications in the Control of Electrical Drives. Závěrečná odborná zpráva projektu GAČR 102/05/2080, VŠB-Technická Univerzita Ostrava, 2007.
- [3] Vas P.: Artificial-Intelligence-Based Electrical Machines and Drives - Application of Fuzzy, Neural, Fuzzy-Neural and Genetic Algorithm Based Techniques. New York: Oxford University Press, 1999.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **prof. Ing. Pavel Brandštetter, CSc.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



doc. Ing. Petr Palacký, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Čestné prohlášení:

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a uvedl jsem veškeré zdroje, které jsem použil při jejím zpracování.

Dne: 1. Května 2013

Podpis autora: 

Poděkování

Chtěl bych poděkovat vedoucímu mé diplomové práce Prof. Ing. Pavlu Brandštetterovi, CSc. Za jeho rady a upomínky, které mi pomohli lépe zvládnout danou problematiku.

ABSTRAKT

Tato diplomová práce se zabývá použitím umělých neuronových sítí v řízení elektrických pohonů. Kdy je řešeno bezsenzorové řízení otáček pomocí estimátoru rychlosti a řízení elektrického pohonu za pomoci prediktivního řízení. Vytváření a trénování jednotlivých neuronových sítí je provedeno v programovém prostředí Matlab- Simulink pomocí Neural Network Toolboxu. Úkolem této práce je teoreticky popsat metody řízení za pomoci neuronových sítí a tyto teoretické poznatky porovnat ze simulačními výsledky.

ABSTRACT

The thesis deals with using of artificial neuron network in control of electrical drives when a sensorless control of rotation by means of the velocity estimator and controlling of electrical drive with predictive controlling is being solved. Generating and training of particular neuron networks is implemented in the programming environment Matlab-Simulink by means of the Neural Network Toolbox. The objective of the thesis is a description of controlling principles by using neuron networks and comparison of the theoretical knowledge and simulation results.

KLÍČOVÉ SLOVA

Umělá neuronová síť, regulace, řízení, regulační obvod, estimátor, prediktivní řízení, elektrický pohon, metoda Backpropagation

KEYWORDS

Artificial neural network, regulativ, driving, controlling systém, estimation, predictive control, electric drive, method of Backpropagation

SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

Σ	suma
BPG	backpropagation metoda (metoda zpětného šíření chyby)
$C\Phi$	konstrukční konstanta
E	střední chyba epochy
f_i	výstup neuronu
H^{-1}	Hessova matice
I_{an}	jmenovitá hodnota proudu motoru
J_m	setrvačná konstanta motoru
J_p	výkonnostní prvek
L_a	indukčnost kotvy
m	velikost epochy
Me	moment kotvy
M_{PM}	moment pracovního mechanismu
N_1, N_2, N_u	definují horizonty, přes které je vyhodnocována chyba
n_n	otáčky motoru
o_j	požadovaná odezva j-tého neuronu výstupní vrstvy daná vzorem trénovací množiny
P_n	jmenovitý výkon motoru
r	žádaná hodnota
R_a	odpor kotvy
T_{ac}	elektromagnetická časová konstanta
T_I	zesílení čidla proudu
T_{TD}	zesílení tachodynamu

T_{Tm}	zesílení tyristorové měniče
u'	pokusný řídicí signál
U_{an}	jmenovitá hodnota napětí motoru
U_{fn}	jmenovitá hodnota napětí budícího obvodu
U_i	indukované napětí
UNS	umělá neuronová síť
w_1, \dots, w_n	váhové koeficienty
x_1, \dots, x_n	vstupy neuronů
x_{pj}	žádaný výstup
\widetilde{x}_{pj}	skutečný výstup z neuronu
y_j	skutečná hodnota odezva j-tého neuronu výstupní vrstvy
y_m	odezva na výstupu UNS
η	koeficient učení
μ	koeficient vlivu změny vah z předchozího kroku (z intervalu $<0,1>$)
ρ	určuje vliv druhého členu vztahu 3.3
θ_i	váhový koeficient z vnějšího prostředí
$\Delta w'_i$	změna synaptické váhy z předchozího kroku

Obsah

SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ.....	6
1.Úvod.....	10
2. Základní poznatky neuronových sítí.	11
2.1 Biologické neuronové sítě.....	11
2.2 Základní struktura umělého neuronu.....	12
2.2 Základní struktura umělé neuronové sítě	15
3. Metody učení neuronových sítí	18
3.1 Metoda Backpropagation	20
3.2 Kwazi-Newtonova metoda.....	21
3.3 Levenberg-Marquardtův algoritmus	21
4. UNS v řízení elektrických pohonů.	23
4.1 Modelování a identifikace dynamických systému neuronovou sítí	23
4.1.1 Identifikace parametrů elektrických pohonů.....	24
4.2 Řízení elektrických pohonů pomocí umělé neuronové sítě.....	24
4.2.1 UNS jako estimátor rychlosti	25
4.2.2 Přímé inverzní řízení	26
4.2.3 Řízení s referenčním modelem.....	27
4.2.4 Řízení s interním modelem	28
4.2.5 Prediktivní řízení	29
4.3 Diagnostika poruch	31
4.4 Pozorování veličin na bázi UNS	33
5. Elektrický pohon	35
5.1 Stejnosměrný motor s cizím buzením	35
5.2 Matematický model.....	36
6. Aplikace metod na příkladu	38
6.1 UNS jako estimátor rychlosti	38
6.2 Prediktivní řízení za pomoci UNS	47

6.3 Zhodnocení použitých UNS	53
7. Závěr.....	54
SEZNAM POUŽITÉ LITERATURY	56
PŘÍLOHY	57

1. Úvod

Oblast ohledně umělých neuronových sítí vznikla, pod záminkou snažit se napodobit lidský mozek a jeho činnosti. Jejich využití, vychází hlavně z jejich základních vlastností, jako přijímat, zpracovat, uchovat, přenášet a využít dané informace. U UNS pak využíváme schopnost učit a řídit, či zpracovávat informace na které UNS nebyla naučena. Neuronové sítě umožňují řešit problémy, kterou jsou klasickými algoritmy jen těžce řešitelné, nebo algoritmické řešení vůbec neznáme

Výhodou UNS je i to, že se dokážou vypořádat i s daty pro které nebyla naučena. Při správném vytvoření neuronové sítě a jejich dobrému natrénování, by měla být neuronová síť v podstatě schopná spočítat jakoukoliv spočitatelnou funkci. Nalézají pak vysoké využití v oblasti zdravotnictví, finančnictví, geologii a řízení.

Tato práce je zaměřena na použití umělých neuronových sítí v řízení elektrických regulovaných pohonů. Bude popsána aplikace a využití u stejnosměrného pohonu. Bude zde popsána a zpracována metoda bezsenzorového řízení otáček pomocí UNS, kdy neuronovou síť použijeme jako estimátor rychlosti a bude zapojen ve zpětné vazbě namísto snímače otáček. Dále bude neuronová síť využita přímo jako regulátor rychlosti podle vybrané metody, jako prediktivní řízení pomocí. Návrh a následná simulace je provedena v programu Matlab-Simulink.

Praktická část je tedy zaměřena na použití UNS v řízení stejnosměrného pohonu, nacházejícího se v laboratoři katedry elektroniky na VŠB-TU. U řízení elektrických pohonů využíváme schopnost UNS řešit problémy v oblasti identifikace a řízení jak lineárních tak a to hlavně, nelineárních regulovaných soustav. Využíváme toho, že dokážou pracovat s libovolným počtem vstupů a výstupů, právě proto jsou vhodné pro identifikace a řízení vícerozměrných dynamických soustav.

Cílem práce tedy je pospat teoreticky oblast využití UNS v řízení elektrických pohonů a následně realizovat vybrané metody na příkladu. Popsat simulační výsledky, porovnat a následně popsát jako se shodují s teoretickým podkladem.

2. Základní poznatky neuronových sítí.

Základ umělých neuronových sítí je založen na poznatcích z oblasti neurofyzologie. Umělé neuronové sítě jsou tedy založeny na základě napodobení neuronových sítí živých organismů. V podstatě se snaží napodobovat funkci lidského mozku, tedy přijímat, zpracovat, uchovat, přenášet a využívat přístupné informace. Avšak nejde o přesnou napodobeninu biologické neuronové sítě, je to jen hrubá napodobenina a umělý neuron je jen přibližný model biologického neuronu. Je totiž zřejmé, že vytvoření umělé neuronové sítě, se všemi schopnostmi biologického mozku je otázkou velmi složitou, ať už z hlediska počtu jeho neuronů, tak jejich propojení apod.

2.1 Biologické neuronové sítě

Uskupení neuronů (nervových buněk), tvoří biologickou neuronovou síť, která je schopna velkého výpočetního výkonu a má ohromnou paměť. Lidský mozek obsahuje přibližně 10^{11} těchto neuronů a asi 10^4 propojení mezi nimi.

Biologický neuron se skládá ze čtyř základních částí:

- Dendrity – což jsou vstupy neuronu. Jsou to krátké vlákna, které přijímají informaci z jiných neuronů
- Soma - tělo buňky, které slouží pro sběr, slučování a transformaci přicházející informace z jiného neuronu
- Axon – je to výstup z neuronu (jedno dlouhé vlákno), které přenáší informaci do dalších neuronů
- Synapse – je to spojovací bod axonu a dendritu z dalšího neuronu (klasický neuron má 10^3 až 10^5 synapsí), která zesiluje nebo zeslabuje vstupní informaci.

Soma (tělo neuronu), provádí shromažďování signálu přijímaných prostřednictvím dendritů. Jestliže úroveň tohoto shromáždění dosáhne vyšší hodnoty než práh excitace neuronu, je takto nashromážděný signál zkonvertovaný, prostřednictvím nelineární aktivační transformace na nervové impulsy poskytující výstup neuronu (aktivace neuronu).

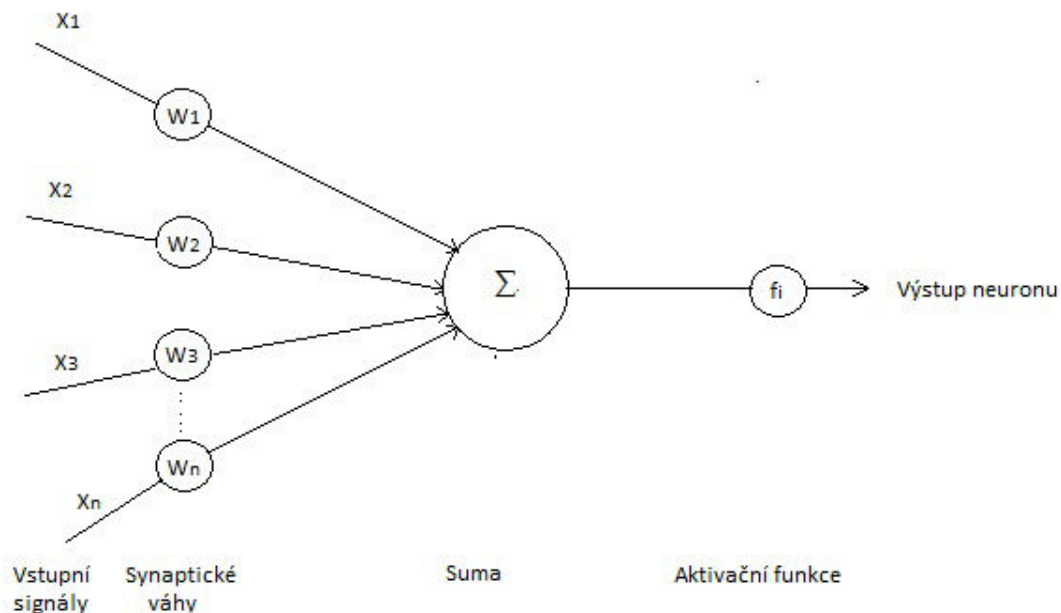
2.2 Základní struktura umělého neuronu

Základním stavebním prvkem každé neuronové sítě je umělý neuron, nazývaný v mnohých literaturách jako také perceptron, což je jeden z nejdůležitějších modelů dodnes používaných, je ovšem nutno zdůraznit, že toto ztotožnění je možné pouze v případě použití speciálního tvaru aktivační funkce (skoková nespojitá funkce).

Model umělého neuronu je uveden na obr. 2.1. Je složen ze vstupů (x_1, \dots, x_n) . Každý z těchto vstupů je vyhodnocen váhovými koeficienty (w_1, \dots, w_n) neboli synaptické váhy, které jsou nastavitelné a potom tyto získané hodnoty sečte v bloku sumátoru Σ . Výsledná hodnota se transformuje ze vstupu na výstup pomocí bloku aktivační funkce neuronu, který informaci předává dál.

Takže shrnutě se umělý neuron skládá z těchto částí:

- *vstupy*
- *váhový sumátor*
- *aktivační funkce (přenosová funkce)*
- *výstupy*



Obr. 2.1. Model umělého neuronu

Výstup neuronu můžeme matematicky popsat takto:

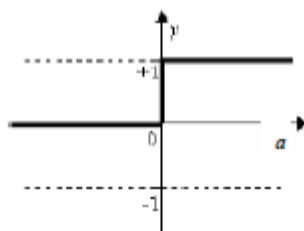
$$y_i = f_i\left[\sum_{j=1}^n w_{ij}x_j + \theta_i\right] \quad (2.1)$$

Kdy θ představuje takzvaný prahový koeficient z vnějšího prostředí.

Aktuální výstup z neuronu je dán zvolenou aktivační funkcí, díky které vkládáme do modelu neuronu nelinearitu. Což umožňuje neuronové síti uchovat si nelineární vztah mezi vstupy a výstupem. Aktivační funkce neuronu, může mít různé tvary. Výběr aktivační funkce je závislý na případě řešené úlohy. Volba aktivační funkce ovlivňuje náročnost technické popř. programové realizace navržené neuronové sítě. Běžně se používají nelineární funkce ve skryté vrstvě a lineární ve výstupní.

Nejčastěji používané aktivační funkce:

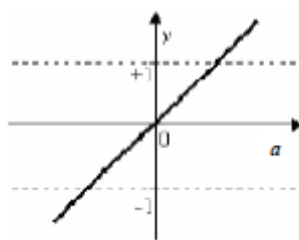
- Skoková aktivační funkce
- Lineární aktivační funkce
- Sigmoidní aktivační funkce
- Hyperbolicko-tangencialní funkce



Obr. 2.2 Skoková aktivační funkce

Vztah aktivační skokové funkce (Obr. 1.2):

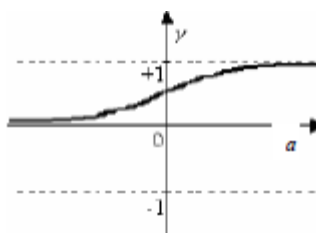
$$y = f(a) = \begin{cases} 1, & a \geq 0 \\ -1, & a < 0 \end{cases} \quad (2.2)$$



Obr.2.3 Lineární skoková funkce

Vztah lineární aktivační funkce:

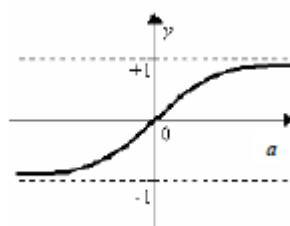
$$y = f(a) = w_0 \cdot a \quad (2.3)$$



Obr. 2.4 Sigmoidní aktivační funkce

Vztah sigmoidní aktivační funkce:

$$y = f(a) = \frac{1}{1+e^a} \quad (2.4)$$



Obr. 2.5 Hyperbolicko sigmoidní funkce

Poslední a nejpoužívanější funkcí je hyperbolicko sigmoidní funkce:

$$y = f(a) = \tanh(a) = \frac{\sinh(a)}{\cosh(a)} = \frac{(e^a - e^{-a})}{(e^a + e^{-a})} = \frac{(e^{2a} - 1)}{(e^{2a} + 1)} \quad (2.5)$$

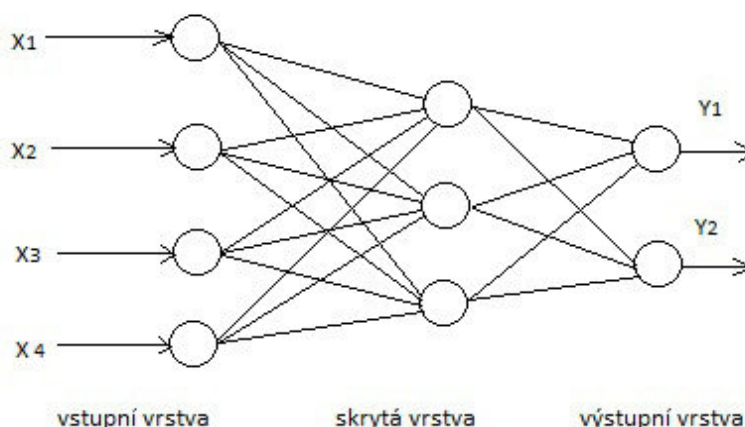
Existuje samozřejmě větší počet aktivačních funkcí různých tvarů (např. radiálního a trojúhelníkového základu), ale ty už jsou méně používané

2.2 Základní struktura umělé neuronové sítě

Struktura neuronové sítě je určena základními prvky sítě a způsobem jejich vzájemného propojení. Neuronová síť většinou představuje vícevrstvou strukturu. Model s úplným propojením spojuje výstup každého neuronu předcházející vrstvy se vstupy všech neuronů následující vrstvy. Částečnou síť nazýváme síť v případě, kdy některé ze vzájemných propojení chybí. [1]

Důležitou vlastností neuronových sítí je schopnost učit se, to znamená schopnost adaptovat se k určité změně z vnějšího okolí a po dobu učení zlepšovat svou výkonnost. Toto učení probíhá na základě určitých trénovacích vzorů. Neuronová síť by měla být tedy schopna reagovat i na neznáme vstupy, na které nebyla naučena.

Zřejmě nejrozšířenějším typem umělé neuronové sítě jsou sítě tzv. vícevrstvé dopředné sítě (feed-forward NN). Počet vrstev i neuronů sítě závisí na dané úloze, kterou síť vykonává. Stavby všech neuronů v síti určují stav neuronové sítě a synaptické váhy všech spojů představují konfiguraci neuronové sítě. V těchto sítích se pohybuje informace jen jedním směrem a to od nejnižší vrstvy (vstupu) do nejvyšší (výstup).



Obr. 2.6 Dopředná vícevrstvá síť s jednou skrytou vrstvou

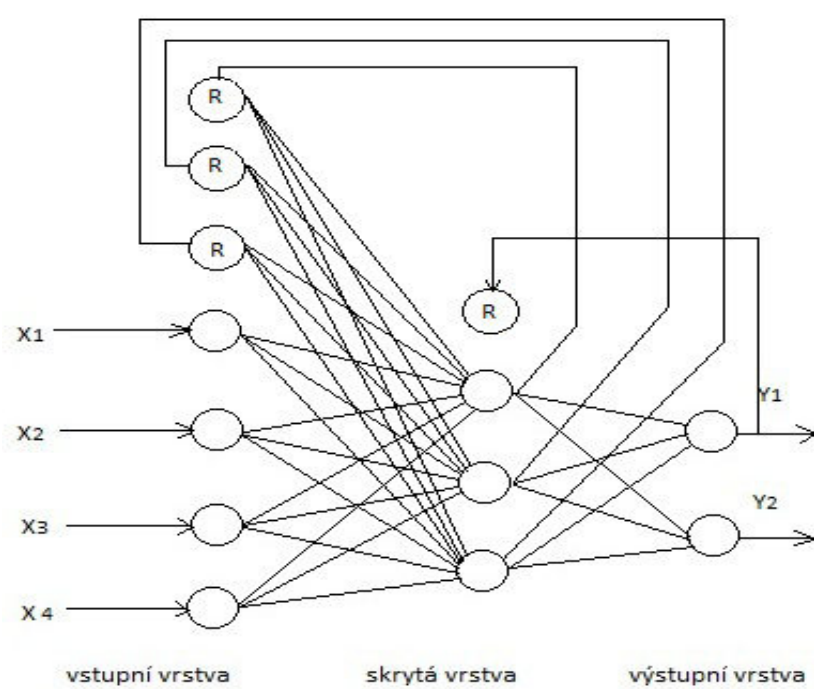
Z obr. 2.6 je zřejmé, že vícevrstvá dopředná síť je řazena do několika vrstev.

Tyto vrstvy rozdělujeme na:

- *vstupní vrstva* (input layer) : neurony dostávají vstup jen z vnějšího prostředí a výstupy z nich většinou pokračují dále do tzv. skryté vrstvy neuronové sítě. V této vrstvě nedochází k žádné úpravě vstupního signálu a slouží jen na oddělení vstupní informace od ostatních vrstev sítě.
- *skrytá vrstva* (hidden layer) : neurony dostávají vstup od ostatních neuronů ale i z vnějšího prostředí přes prahové připojení a jejich výstupu pokračují dále do neuronové sítě. Hodnotu aktivity skryté vrstvy určují aktivity vstupních neuronů spolu se synaptickými váhami na spojení mezi vstupní a skrytou vrstvou, a taky vstupy z vnějšího prostředí
- *výstupní vrstva* (output layer) : je podobná skryté vrstvě, ale její výstupy z ní jdou vnějšího prostředí. Chování výstupních neuronů je závislé na aktivitách skrytých neuronů a synaptických váh na spojení skryté vrstvy a výstupní vrstvy, spolu se vstupy z vnějšího prostředí.

Na tomto základě pak dělíme i neurony na vstupní, skryté a výstupní

Při modelování dynamických systémů se hlavně používají tzv. rekurentní dopředné sítě, díky kterým lze do vícevrstvé sítě implementovat časový kontext. Tato síť využívá zpětné vazby k šíření signálu, ne jen směrem z vrstvy vstupní do výstupní, ale dochází i ke zpětnovazebnímu přenosu informací z vrstev nižších zpět do vrstev vyšších. Díky této zpětné vazbě, zvyšujeme významně účinnost učení a výkonnost sítě. Jsou dynamické a jejich stav se nemění, dokud nedosáhnou bodu rovnováhy, následně zůstanou v tomto bodě, dokud se nezmění vstup a není nutné hledat nový rovnovážný stav.



Obr. 2.7 Rekurentní dopředná síť.

3. Metody učení neuronových sítí

Jak již bylo naznačeno v předcházející kapitole, nejvýznamnější vlastností umělé neuronové sítě je schopnost učit se. Existuje mnoho algoritmů učení umělých neuronových sítí, kterou jsou charakteristické jak pro daný typ sítě tak metodou učení.

Známe tři základní skupiny, díky kterým dělíme učící algoritmy, dle učícího procesu. Tyto skupiny jsou:

- *Učení s učitelem* – je to kontrolované učení, které se většinou uskutečňuje dle vhodných vzorů. Neuronové síti je předkládána množina příkladů vhodného chování sítě – vstupů a odpovídajících výstupů. Cílem procesu je aby neuronová síť, na základě explicitní informace, které představují vzorové páry vstupů a výstupů, dosáhly takové konfigurace, na základě které dokáže realizovat předepsané zobrazení. Tento proces se často využívá pro identifikaci, modelování a diagnostiku systému.
- *Učení bez učitele* – je to takzvané nekontrolované učení tzn., že pro správnou funkci UNS stačí zadat pouze vstupní vektory a UNS na základě svých interních pravidel sama vyhledá charakteristické znaky procesu, a podle nich upraví své nastavení. Tato metoda je časově náročnější než předešlá a používá se především u Kohonenových a samoorganizujících sítí.
- *Známkové učení* – je to proces nacházející se na rozhraní předcházejících procesů. Při tomto učení, učitel poskytuje neuronové síti hodnocení její reakce na přicházející vstupy tak, aby proces učení vedl k žádanému cíli. Tento proces má významnou úlohu v oblasti řízení.[1]

Proces trénování:

- začíná inicializací všech prahových a váhových koeficientů malým nenulovým číslem. Většinou jsou tyto váhové koeficienty generovány náhodně, ale používají se i genetické algoritmy.
- nadále se postupně na vstup sítě předkládá část, nebo celá trénovací množina, u čehož se zaznamenává chyba odhadu sítě
- potom se znova nastaví váhy, tak aby se znova snížila chyba odhadu sítě
- tento proces se dále opakuje dokola, dokud chyba neklesne na hodnotu minimální žádané chyby.

Jeden takový cyklus se nazývá epocha, kdy velikostí trénovací množiny se rozumí počet vzorků vstupně-výstupních dat, které byly použity pro nastavení vah a nazýváme ji velikost epochy.

Na aktivaci výstupních neuronů je nejpoužívanější metoda střední kvadratické chyby (MSE). Je lehce vypočitatelná a její parciální derivace podle jednotlivých prahových koeficientů se dají určit explicitně. V praxi se tato metoda dobře osvědčila. Střední kvadratická chyba trénovacích vzorků, se určí jako kvadrát odchylky mezi hodnotou žádaného výstupu j -tého výstupního neuronu. Nakonec se spočítá jejich průměrko všechny výstupní neurony sítě.

$$E_p = \frac{1}{n} \sum_{j=1}^n (x_{pj} - \widetilde{x}_{pj})^2 \quad (3.1)$$

kde n je počet výstupních neuronů, x_{pj} je žádaný výstup j -tého neuronu \widetilde{x}_{pj} skutečný výstup neuronu. Pro střední chybu epochy platí:

$$E = \frac{1}{m} \sum_{p=1}^m E_p \quad (3.2)$$

Kde m je velikost epochy.

Trénování sítě se může být ukončeno při dosažení maximálního počtu epoch, kterých má dosáhnout během adaptace sítě, nebo při dosažení minimální chyby epochy.

Dobře natrénovaná síť je schopná predikovat správné hodnoty výstupu i pro neznámé vstupy. Každý nový vstup vede k takové hodnotě výstupu, která je blízká hodnotě příslušné k vstupu trénovacích dat. To znamená, že síť bude vydávat dobré výsledky bez toho, aby byla trénována na všechny možné vstupně- výstupní data. [2]

3.1 Metoda Backpropagation

Metoda *backpropagation neboli metoda* zpětného šíření byla vytvořena jako učící algoritmus pro učení vícevrstvých neuronových sítí s učitelem a byla vybrána jako přelomový princip učení a je v současné době nejpobulárnější. V podstatě se jedná o to, že tato metoda nastavuje váhy jednotlivých spojů zpětným šířením chyby přes síť tak, aby velikosti nastavených biasů byly z hlediska řešeného problému optimální. Obecný tvar změny váhových koeficientů můžeme matematicky vyjádřit:

$$w_{ij}(k + 1) = w_{ij}(k) + \Delta w_{ij}(k), \quad (3.3)$$

kde index i označuje uvažovaný neuron a index j neuron na něho působící z předcházející vrstvy.

Avšak u této metody probíhá nastavování vah opačným směrem, než jakým šíří vstupní informace. U tohoto algoritmu rozeznáváme dvě fáze: aktivační a adaptační.

Aktivační funkce se používá k inicializaci vah, což se obvykle dělá pomocí generátoru náhodných čísel. Kromě použití, generátoru náhodných čísel existují i jiné metody pro počáteční nastavení vah jako simulované žihání a genetické algoritmy. U této fáze se informace šíří ze vstupu na výstup. Při adaptační fázi je výstupní vektor porovnáván s námi požadovaným originálem a rozdíl mezi oběma vektory je použit pro výpočet nových vah tak, že se nejprve opraví váhy výstupní vrstvy.[2]

Podstata metody backpropagation spočívá v nalezení minima chybové funkce E , kterou můžeme např. definovat takto:

$$E = \frac{1}{2} \sum_{i=1}^p \sum_{j=1}^m (y_j - o_j)_i^2, \quad (3.4)$$

kde y_j skutečná hodnota odezva j -tého neuronu výstupní vrstvy
 o_j požadovaná odezva j -tého neuronu výstupní vrstvy daná vzorem trénovací množiny

- p celkový počet vzorů trénovací množiny
m počet neuronů výstupní vrstvy.

Změna či úprava synaptických vah mezi neurony i a j je cesta, jak dosáhnout cíle v nalezení minima chybové funkce. Tato změna je dána vztahem

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} + \mu \Delta w_i', \quad (3.5)$$

- kde η koeficient učení
 μ koeficient vlivu změny vah z předchozího kroku (z intervalu $\langle 0,1 \rangle$)
 $\Delta w_i'$ změna synaptické váhy z předchozího kroku

Výpočet korekce pro změnu synaptické váhy je přímo úměrná parciální derivaci $\partial E / \partial w_i$

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial a} \cdot \frac{da}{dx} \cdot \frac{\partial x}{\partial w_i} + \frac{\partial E}{\partial b} \cdot \frac{db}{dy} \cdot \frac{\partial y}{\partial w_i} \quad (3.6)$$

3.2 Kwazi-Newtonova metoda

Tato metoda využívá Newtonův optimalizační algoritmus, jehož základní vztah je:

$$w(k) = -H^{-1}(k) \cdot g(k)$$

Kwazi-Newtonova metoda používá aproximaci vyššího řádu, oproti metodě BPG, která využívá aproximaci kritériální funkce v okolí pracovního bodu $w(k)$. Nevýhodou této metody je náročnost na výpočet Hessovy matice H^{-1} . Určení Hessovy H^{-1} matice je možné totiž jen v případě je-li matice H regulární a to ve většině případů nebývá splněno.

3.3 Levenberg-Marquardtův algoritmus

Jednou z modifikací metody backpropagation je právě Levenberg-Marquardtův algoritmus, který velmi často používá právě u vícevrstevných dopředných sítí. Jedná se v podstatě o kombinaci Kwazi-Newtonovy metody a metody Backpropagation. Je velmi vhodný k trénování rozsáhlejších sítí díky jeho velmi rychle konvergence. To se nejvíce projeví v případech, kdy je nutná vysoká přesnost

učení. Algoritmus předpokládá účelovou funkci ve tvaru čtverců, což je typické pro trénování dopředných neuronových sítí.

Algoritmus učení lze vyjádřit hodnotou změny vah spojení:

$$\Delta w(k) = -[J^T(k)J(k) + \alpha I(k)]^{-1}g(k) \quad (3.7)$$

Levenbergův-Marquardtův algoritmus přechází na kvazi-Newtonovu metodu, jestliže se hodnota α blíží k nule, ale pokud je α velké, blíží se BPG algoritmu s nízkou hodnotou učení. Parametr α se adaptivně mění tak, že po každém úspěšném kroku, což znamená, že se kriteriální funkce sníží na hodnotu parametru α a zvýší se za předpokladu, že byla kriteriální funkce v aktuální iteraci zvýšena.

[2]

4. UNS v řízení elektrických pohonů.

V této kapitole budou popsány základní možnosti využití umělých neuronových sítí v řízení elektrických pohonů. Stručně můžeme použití UNS v této oblasti shrnout do pár základních bodů:

- modelování systému
- identifikaci parametrů systému
- řízení pohonů
- diagnostiky poruch
- pozorování stavových veličin

Tato práce bude především směřována k estivaci a regulaci neboli sledování veličin elektrického regulačního motoru. Ale přeci jen zde bude alespoň stručně popsána aplikace umělé neuronové sítě i v jiných uplatněních v elektrických regulačních pohonech.

4.1 Modelování a identifikace dynamických systému neuronovou sítí

Základní úlohou identifikace je složení matematických modelů zkoumaných objektů pro daný cíl, kterým může být poznání zákonitosti zkoumaného objektu, nebo jeho řízení. Matematicko – fyzikální analýza a experimentální identifikace identifikovaného objektu jsou dvě základní metody, které se využívají při tvorbě identifikačních modelů. Současná praxe dokazuje, že identifikace složitých objektů si vyžaduje kombinaci obou těchto metod, přičemž matematická analýza fyzikální podstaty zkoumaných objektů byla a bude základní metodou úspěšné strukturální identifikace zkoumaného objektu. [1]

4.1.1 Identifikace parametrů elektrických pohonů

Měření odezvy motoru na testovací signál je základem identifikace parametru elektrického pohonu. Automatická identifikace má v dnešní výpočetní době řadu výhod a to např.:

- možnost generování složitých testovacích signálů
- sofistikované identifikační algoritmy
- možnost on-line identifikace
- průběžné využití výsledků identifikace pro zkvalitnění řízení pohonů

Pro návrh struktury identifikace parametrů pohonů využívající UNS se většinou používají dva způsoby. Kdy u jednoho, je identifikace parametru je založena na principu aproximace závislosti hledaného parametru na měřitelných veličinách. Tento návrh se většinou provádí při identifikaci založených na rozsáhlých UNS v režimu off-line. Ovšem v režimu off-line trénování je problém s výběrem vhodných testovacích signálů pro trénovací algoritmus neuronového identifikátoru. Z toho vyplývá, že trénovací algoritmus může být oproti druhému přístupu složitější a časově náročnější.

U druhého způsobu se hodnoty parametrů získávají na základě analogie struktury UNS se strukturou identifikovaného elektrického pohonu, kdy se UNS využívá jako paralelní nebo serio-paralelní model systému. V tomto případě má UNS většinou velmi jednoduchou strukturu s malým počtem neuronů. Použitý trénovací algoritmus je velmi rychlý a hodnoty identifikovaných parametrů většinou odpovídají přímo parametrům UNS, ale můžeme je získat taky nepřímým způsobem a to výpočtem z adaptivních parametrů sítě. [1]

4.2 Řízení elektrických pohonů pomocí umělé neuronové sítě.

V této kapitole bude popsáno pár základních možností řízení elektrického pohonu pomocí UNS. Jelikož v této práci budou návrhy a simulace prováděné na stejnosměrném motoru s cizím buzením, budou zde popsány hlavně metody pro řízení ss pohonu.

UNS pro řízení elektrických pohonů se používají především dvěma způsoby a to buď jako model nebo regulátor systému. V regulačních systémech můžeme neuronové regulátory rozdělit do čtyř základních skupin: přímé inverzní řízení (direct inverse control), řízení s referenčním modelem

(model reference control), řízení s interním modelem (internal model control), prediktivní řízení (predictive control).

4.2.1 UNS jako estimátor rychlosti

UNS jako estimátor rychlosti se využívá u problematiky bezsenzorových pohonů, kde je možné využívat jak neuronové sítě, tak fuzzy-neuronové sítě.

Tyto sítě lze použít i při minimální znalosti parametrů elektrického pohonu, a lze docílit minimální závislosti přesnosti estimace rychlosti na měnících se parametrech motoru. Tento fakt je výhodou především vůči otevřeným estimátorům využívajícím matematický model stroje.

Jelikož není doposud nalezen přesný algoritmus pro určení struktury sítě pro danou aplikaci musíme síť získat pomocí testování různých topologií a zjišťování přesnosti aplikace. Pro realizaci estimátoru je nutné především určit vhodnou strukturu neuronové sítě, aby bylo dosaženo co nejjednodušší neuronové sítě s dostatečnou přesností odhadu, což je základní cíl pro využití UNS v praxi.

Hlavní důvody nasazení bezsenzorového pohonu jsou:

- *redukce ceny a použitého hardwaru*
- *nárůst mechanické robustnosti*
- *nasazení v agresivním prostředí*
- *vyšší spolehlivost*
- *snížení požadavku na údržbu*
- *nárůst šumové imunity*

4.2.1.1 Pozorovatelé- Kalmanův filtr a Luendbergův algoritmus

V ustálených a přechodových stavech je pro zlepšení vlastností elektrického pohonu možné použít pozorovatele. Jedná se o skupinu uzavřených estimátorů, kde dochází k menší závislosti estimovaných veličin na parametrech motoru. V případě použití v nelineárních systémech se aplikuje rozšířený Kalmanův filtr. Jedná se o rekurzivní filtr, který je založen na znalosti statistik stavů a šumů vytvořených měřeními a systémovým modelováním.

Deterministický rozšířený Luendbergův pozorovatel je rovněž jednou z možností pro nasazení při řešení otázek řízení elektrického pohonu. Nevýhodou těchto pozorovatelů je nutná znalost

matematických modelů motorů, která v případě umělé inteligence odpadá. Další nevýhodou je nutná znalost kovariačních matic, tedy jejich nastavování a filtrace.

4.2.2 Přímé inverzní řízení

Tuto metodu můžeme považovat za intuitivně nejjednodušší případ neuronového regulátoru. Použití UNS je zaměřeno jako inverzní model systému. U tohoto systému jsou známy případy s problémy ohledně robustnosti a stability, jelikož se zde nenachází zpětná vazba. I přesto jsou tyto modely nejpoužívanější, jelikož se u mnoha procesů, problém s inverzí nevyskytuje.

Předpokládáme-li, že inverzní model píšeme rovnicí:

$$y(k+1) = g(y(k), \dots, y(k-n+1), u(k), \dots, u(k-m)) \quad (4.1)$$

Pak neuronovou síť naučíme jako inverzi soustavy:

$$u(k+1) = g^{-1}(y(k+1), \dots, y(k-n+1), u(k), \dots, u(k-m)) \quad (4.2)$$

Pak se připojí inverzní model k soustavě tak, že na místo výstupu $y(k+1)$ přivedeme žádanou hodnotu $r(k+1)$.

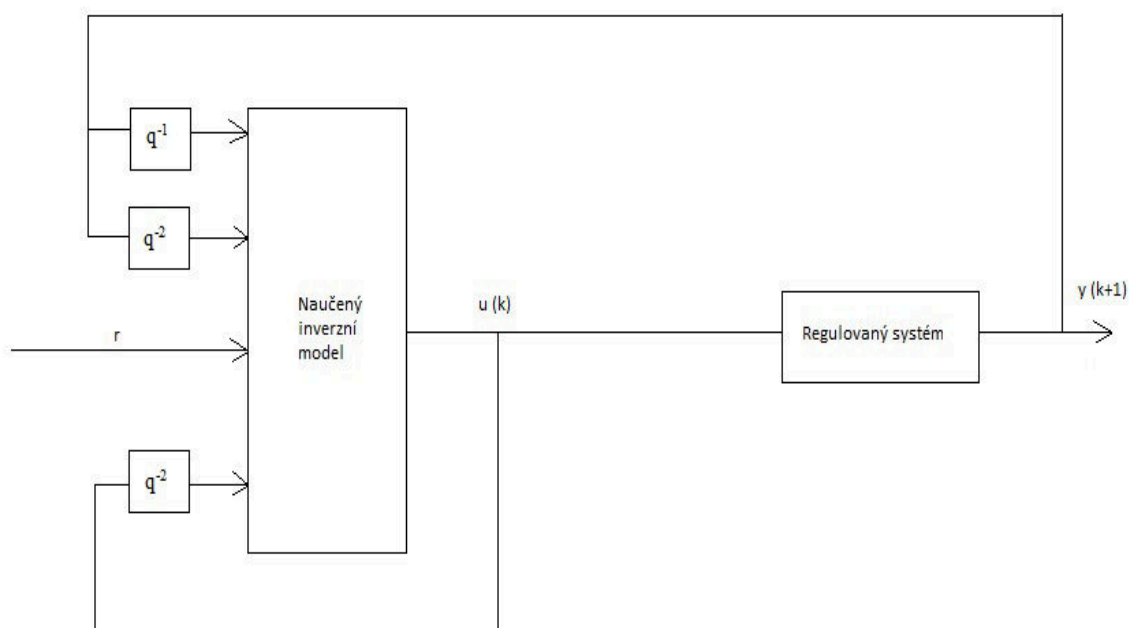
Základní vlastnosti metody:

Hlavní výhody:

- Intuitivně jednoduchá
- Velice jednoduchá na implementaci
- Vhodná pro aplikaci on-line trénování pro časově proměnné systémy

Hlavní nevýhody:

- Nelze použít v případě, kdy inverzní model regulovaného systému neexistuje
- Problémy s inverzními modely, které nejsou dobře tlumeny [7]



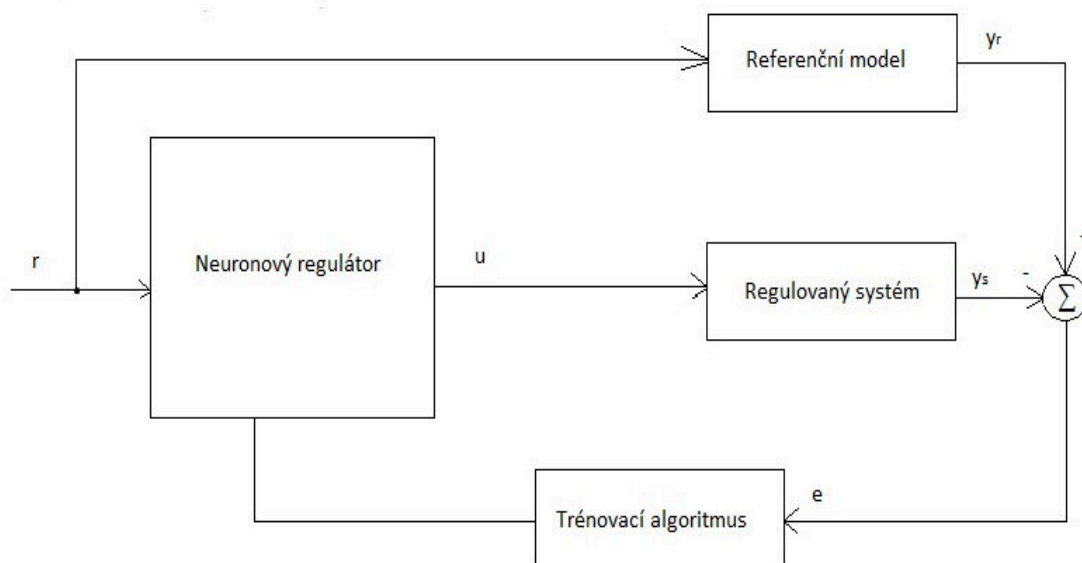
Obr. 4.1 Přímé inverzní řízení

Z obrázku 3.1 vidíme, že vstupní signály neuronové sítě jsou dvě předchozí hodnoty výstupu regulovaného procesu (úhlová rychlost ss pohonu), potom předchozí hodnota z výstupu neuronové sítě a taky požadovaná hodnota r (úhlová rychlost).

Inverzní modely se používají i v kombinaci s klasickými regulátory s pevnými parametry. V takovém zapojení zabezpečuje klasický regulátor řízení v ustálené oblasti a při dynamických procesech se projevuje účinek UNS regulátoru.

4.2.3 Řízení s referenčním modelem

Úlohou řídicího systému tvořenou neuronovou sítí je zabezpečit, aby výstup ze systému y_s sledoval výstup referenčního modelu y_r . Na trénování neuronové sítě je použit právě rozdíl těchto hodnot. Oproti přímému inverznímu řízení, struktura s referenčním modelem řeší právě problém s robustností a máme možnost adaptace neuronového regulátoru, který je trénovaný on-line.



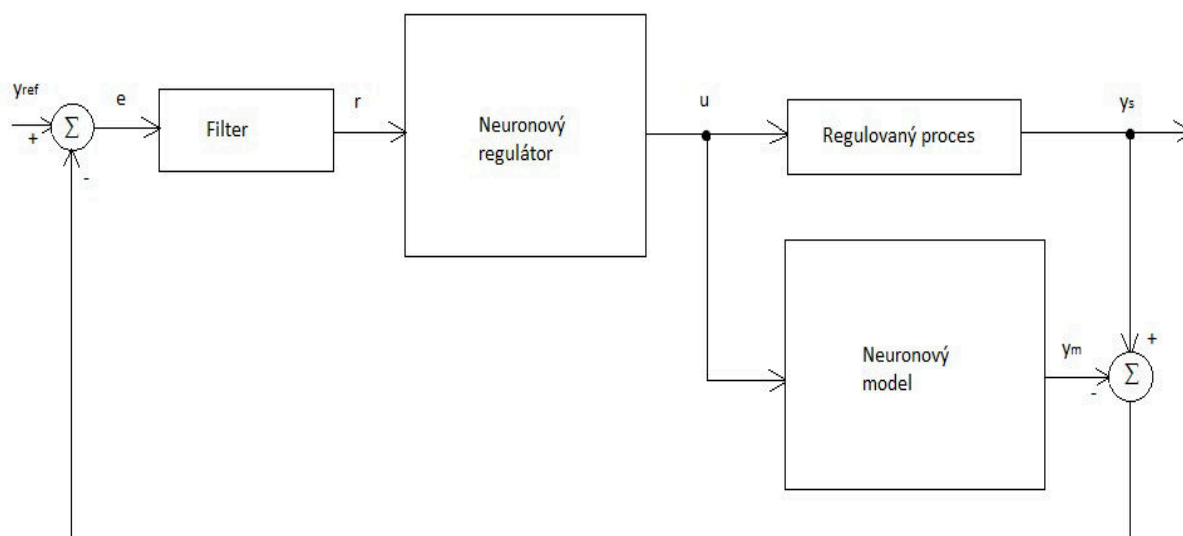
Obr. 4.2 Řízení s referenčním modelem

4.2.4 Řízení s interním modelem

U tohoto řízení se využívají dva neuronové modely. Obě tyto sítě jsou použity v řídicí struktuře. Jedna síť je použita jako neuronový regulátor a druhá síť jako dopředný model soustavy. V této struktuře řízení je neuronový model systému zapojen paralelně s reálným systémem.

Zpětnovazební signál je tvořen rozdílem jejich výstupů, který zpracovává regulátor tvořený zpětnovazebním inverzním modelem systému.

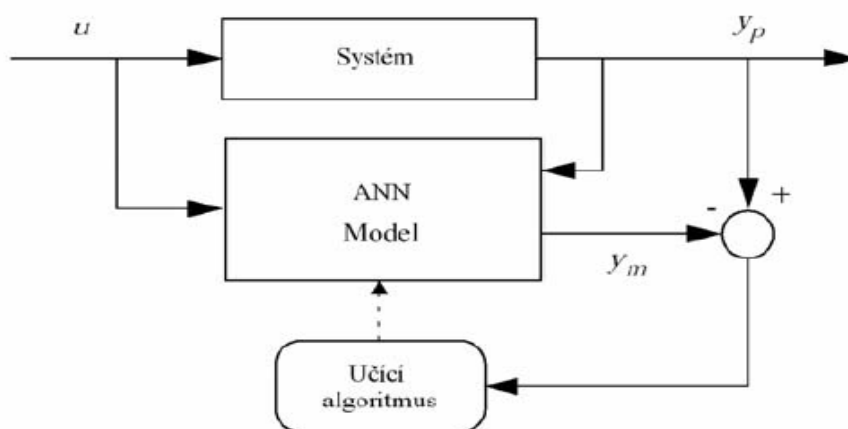
V řídicí smyčce tak vlastně vzniká integrátor, díky čemuž toto zapojení reguluje na nulovou ustálenou odchylku při konstantní žádané hodnotě. Konstantní porucha bude také vyregulovaná, protože bude zahrnuta do neuronového modelu. Jinou možností, jak zajistit nulovou ustálenou odchylku i při jiných regulátorech, je zapojení PI regulátoru do vnější smyčky. [3]



Obr. 4.3 Řízení s interním modelem

4.2.5 Prediktivní řízení

Před samotným řízením regulované soustavy UNS, je zapotřebí systém identifikovat. Při identifikaci tohoto systému je nutné vytvořit na neuronové síti založit potřebný model, který chceme regulovat. Model systému je většinou trénován off-line Levenberg-Marquardtovou metodou.



Obr. 4.4 Blokové schéma inicializace systému.

Metoda prediktivního řízení je založena na technice ustupujícího horizontu (receding horizon technique). Založený model na neuronové síti predikuje odezvu systému přes specifikovaný časový

horizont. Minimalizaci uvedeného výkonnostního kritéria přes zadaný horizont je dán řídicím signálem, který je dán optimalizačním programem.

$$J_p = \sum_{j=N_1}^{N_2} (r(k+j) - y_m(k+j))^2 + \rho \sum_{j=1}^{N_u} (u'(k+j-1) - u'(k+j-2))^2 \quad (4.3)$$

Kde parametry:

$N_1, N_2, N_u \dots$ definují horizonty, přes které je vyhodnocována chyba

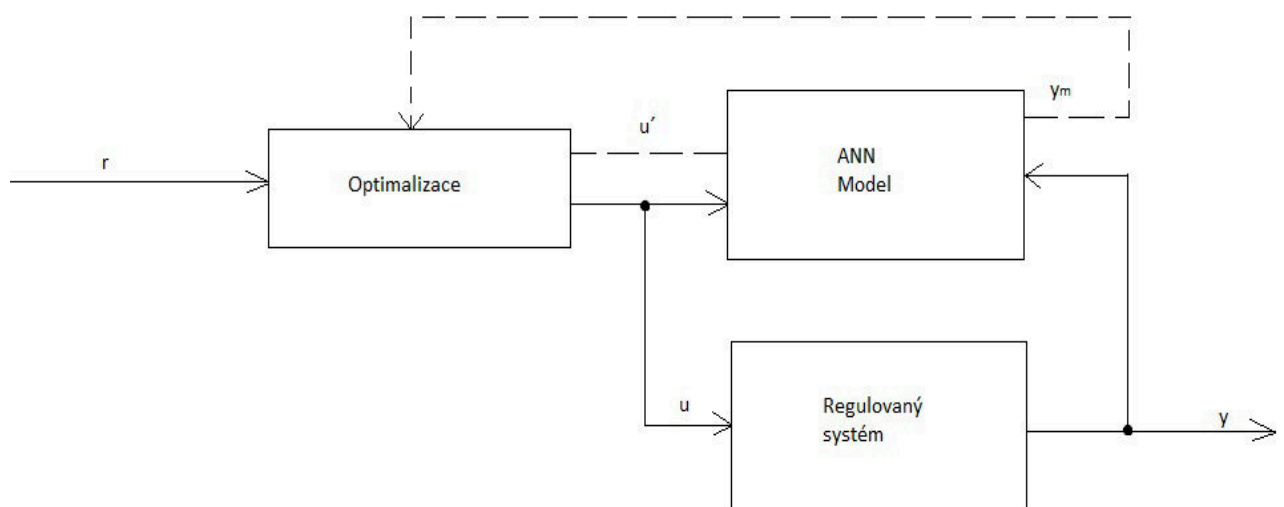
r žádaná hodnota

y_m odezva na výstupu UNS

u' pokusný řídicí signál

ρ určuje vliv druhého členu vztahu 3.3

J_p výkonnostní prvek



Obr. 4.4 Prediktivní řízení

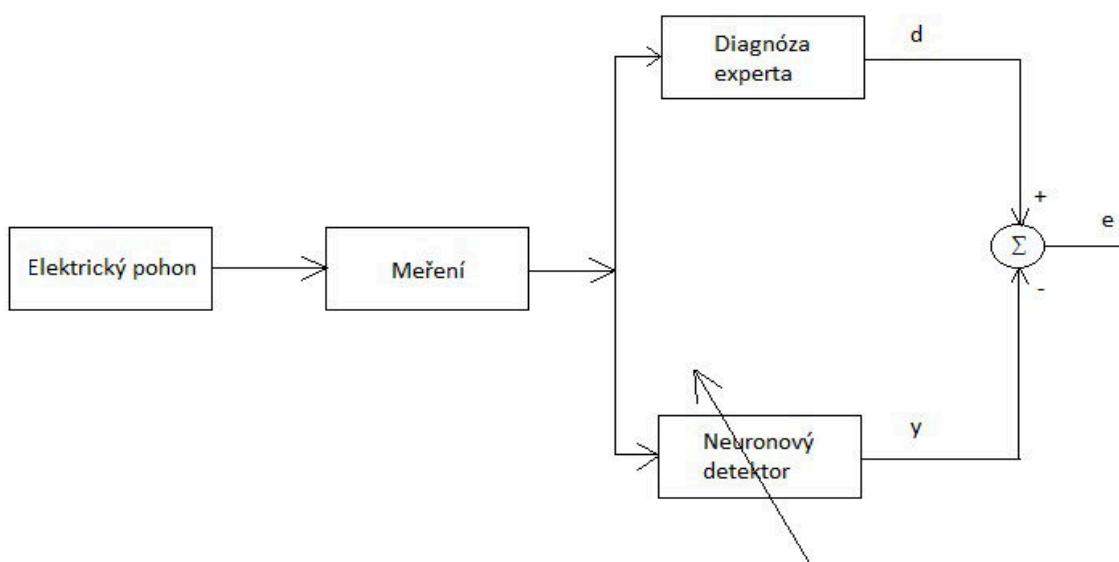
Na obrázku 4.4 je blokové schéma prediktivního řízení využívající UNS. Regulátor se skládá z ANN modelu a optimalizačního bloku. Optimalizační blok určuje hodnotu u' , která minimalizuje výkonnostní kritérium J_p .

4.3 Diagnostika poruch

Normálně se pro komplexní diagnostiku používají expertní systémy. Ten na základě údajů o stavu technologického procesu spolu s informacemi operátora, určuje podle pravidel v bázi znalostí nejpravděpodobnější příčiny havarijního stavu procesu. Expertní systém je doplněn o editor, kterým je možné měnit pravidla a rozšiřovat bázi znalostí o další poznatky. Při hledání příčiny poruchy expertní systém načítá stav zařízení z řídicího počítače. Potom formou jednoduchých otázek, které dává operátorovi se snaží získat další informace. Takhle získané informace jsou nedosahitelné normálním měřením a sledováním procesu.

Vývoj diagnostiky konvertuje k různým stupňům využití metod umělé inteligence. V oblasti diagnostiky za pomoci použití umělé inteligence se nejvíce prosazovaly expertní systémy, ale v současnosti se stále více rozšiřují metody na bázi umělých neuronových sítí a fuzzy logiky.

Umělé neuronové sítě jsou v oblasti diagnostiky a predikci poruch většinou trénovány na expertní bázi vědomostí, tj. množině údajů charakterizující stavy v různých pracovních režimech. Tyto údaje mohou být zobrazené během měření na reálném systému nebo prostřednictvím simulace na matematických modelech.



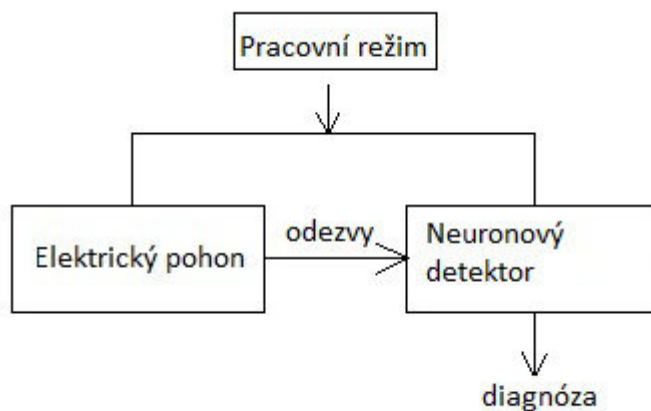
Obr. 4.5 Proces učení neuronové detektoru.

Proces učení neuronového detektoru, který je vyobrazen na obr.4.5., probíhá většinou v off-line režimu. Z obrázku je vidět, že při off-line učení jsou parametry nastavované na principu rozdílu

rozhodnutí experta a neuronového detektoru. Takto nastavený detektor pracuje v podstatě jako klasifikátor stavů zařízení.

Podle testovacích signálů, rozlišujeme dva případy diagnostiky, ve kterých může neuronový detektor pracovat:

- Stimulační diagnostiky, kdy se na diagnostikovaný objekt připojují vybrané testovací signály
- Funkcionální diagnostika (obr. 4.6), při které diagnostikovaný objekt reaguje na vstupů signály vytvořené v pracovním režimu řízení.[1]



Obr. 4.6 Neuronový detektor pro funkcionální diagnostiku

4.4 Pozorování veličin na bázi UNS

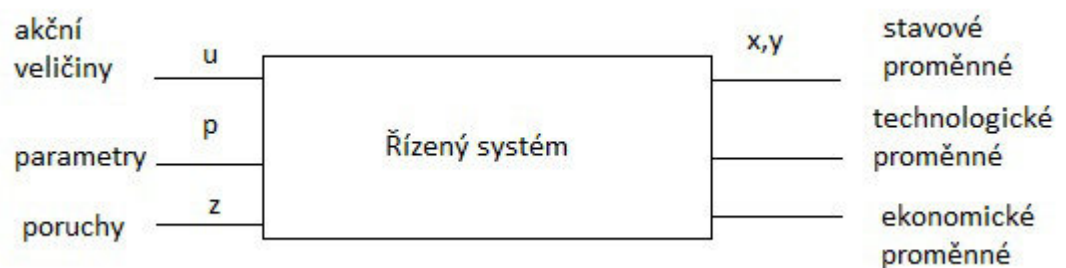
V technologických procesech existuje velký počet vstupních a výstupních veličin, které jsou měřitelné, ale i které měřitelné nejsou.

Vstupní veličiny systému, můžeme rozdělit do dvou skupin:

- Veličiny, které můžeme využít pro řízení procesu, tzn. Jsou to řídicí proměnné u nebo konstantní parametry procesu p , které ovlivňují chování procesu
- Veličiny, které není možné přímo využít pro řízení- např. vektor poruchových veličin z , které můžou a nemusí být měřitelné a často mají náhodný charakter.

Podobně můžeme rozdělit i výstupní proměnné do dvou skupin

- Veličiny, které jsou důležité z hlediska řízení procesu – jsou to nejčastěji stavové proměnné x , a nebo lineární a nelineární funkce stavových proměnných y .
- Veličiny, které jsou důležité ne z hlediska řízení, ale z hlediska technologie a ekonomiky procesu. Jsou to veličiny, které jsou důležité pro sestavení účelových funkcí při optimalizaci a které, když se dají vyjádřit prostřednictvím proměnných x , vystupují jako funkce stavových veličin. [1]



Obr. 4.7 Vstupní a výstupní veličiny řízeného procesu

Pozorovatele veličin na bázi UNS dělíme podle různých kritérií. Podle veličiny, které rekonstruují je dělíme na:

- pozorovatele stavu x
- pozorovatele výstupních signálu y
- pozorovatele parametrů p
- pozorovatele stavu a parametru x, p
- pozorovatele poruchy z

Při pozorování stavových a výstupních veličin systémů, může do popředí vstupovat UNS v různých strukturách, jenž vybíráme podle řešení daného problému. Nejčastěji se UNS pro pozorování veličin používají k úlohám jako: aproximace, predikce, asociace, klasifikace, řízení a oddělení zdrojů ovšem hlavně využíváme znalosti z řešení úloh řízení a predikce.

1. Úloha aproximace

Jestliže je výstupní obraz nelineární vstupně-výstupní popsán vztahem:

$$d = g(x) \quad (4.4)$$

Kde vektor x je vstupem a skalár d výstupem, potom úlohou je navrhnout UNS, která bude aproximovat neznámou nelineární funkci g , přičemž pro trénovací proces jsou zadány vstupně-výstupní vzory (x_i, a_i) . V tomto případě používáme kontrolované učení.

2. Úloha predikce

U této úlohy máme obvykle k dispozici m předcházejících vzorků vstupů $x(k-1), x(k-2), \dots, x(k-m)$. Úlohou je navrhnout UNS, která bude na základě vstupů z předcházejících vzorků predikovat následující vzorek $x(k)$, tzn. Sestavit model stochastického procesu generující časovou posloupnost vzorků.

Chybu predikce můžeme vyjádřit ve tvaru

$$e(k) = x(k) - x_N(k), \quad (4.5)$$

kde $x_N(k)$ označuje výstup sítě.

Takže cílem predikce je předpovídat krátkodobý vývoj chování systému. Schopnost účinně sledovat stav systému je určena kvalitou pozorování systému

5. Elektrický pohon

Tato práce je zaměřena na aplikaci umělých neuronových sítí v řízení elektrických pohonů. Umělá neuronová síť v této práci je navrhovaná pro stejnosměrný motor s cizím buzením, nacházejícím se v laboratoři katedry elektroniky VŠB-TU Ostrava. V této kapitole bude tento motor popsán a vytvořen jeho matematický model, který nám usnadní práci v programu Matlab-SIMULINK.

5.1 Stejnosměrný motor s cizím buzením

V úvodu kapitoly již bylo naznačeno, že se bude pracovat se stejnosměrným motor s cizím buzením (permanentní magnety) jenž má následující parametry:

- $U_{an} = 440 \text{ V}$,
- $I_{an} = 37,5 \text{ A}$,
- $P_n = 15 \text{ kW}$,
- $n_n = 28 - 2800 \text{ ot/min}$,
- $J_m = 0,24 \text{ kgm}^2$,
- $CB I_{fn} = 1 \text{ A}$,
- $U_{fn} = 190 \text{ V}$,
- $c\Phi = 1,42 \text{ Vs}$,
- $R_a = 0,616 \Omega$,
- $L_a = 13,18 \text{ mH}$.

Elektrický pohon je napájen šestipulzním tyristorovým měničem SIMOREG 6RA24 se zesílením $K_{TM} = 5,4$ a jeho časová konstanta je $1,67 \text{ ms}$.

U pohonu jsou požitá čidla pro snímání proudu a otáček o následujících parametrech:

Čidlo proudu: zesílení $K_I = 0,333 \text{ V/A}$, časová konstanta $T_i = 1 \text{ ms}$.

Čidlo otáček (tachodynamo) : $K_{TD} = 0,095 \text{ V/(rad/s)}$, časová konstanta $T_{TD} = 1 \text{ ms}$.

Moment setrvačnosti uvedeného pohonu je $J_c = 0,3 \text{ kgm}^2$

Elektromagnetická časová konstanta:

$$T_{ac} = \frac{L_{ac}}{R_{ac}} \quad (5.1)$$

Po dosazení hodnot vyšla elektromagnetická časová konstanta $T_{ac} = 21 \text{ ms}$.

Elektromechanická časová konstanta:

$$T_m = \frac{J_c R_{ac}}{(c\Phi)^2} \quad (5.2)$$

Po dosazení hodnot vyšla elektromechanická časová konstanta $T_m = 91 \text{ ms}$.

Jelikož u tohoto pohonu platí, že $T_m > 4 T_{ac}$, budeme zanedbávat změny indukovaného napětí, jelikož probíhají pomaleji, než přechodný děj proudu v kotvě.[3]

5.2 Matematický model

Výhodou stejnosměrných motorů je to, že se dají popsat docela jednoduchým matematickým modelem. Nemůžeme však předpokládat, že tento model je absolutně přesný, jde spíše o přibližnou představu. Nám to však pro řešení regulace vystačuje.

Obvod kotvy může popsat následujícími rovnicemi:

$$u_a = R_{ac} + L_{ac} \frac{di_a}{dt} + u_i \quad (5.3)$$

$$u_i = c\Phi\omega \quad (5.4)$$

Závislost momentu kotvy:

$$M_e = c\Phi i_a = J_c \frac{d\omega}{dt} + M_L \quad (5.5)$$

V regulačních elektrických pohonech se však používá především přechod na operátorové přenosy pomocí Laplaceovy transformace, která nám umožňuje složité derivace a integrály nahradit Laplaceovou transformátorovou složkou.

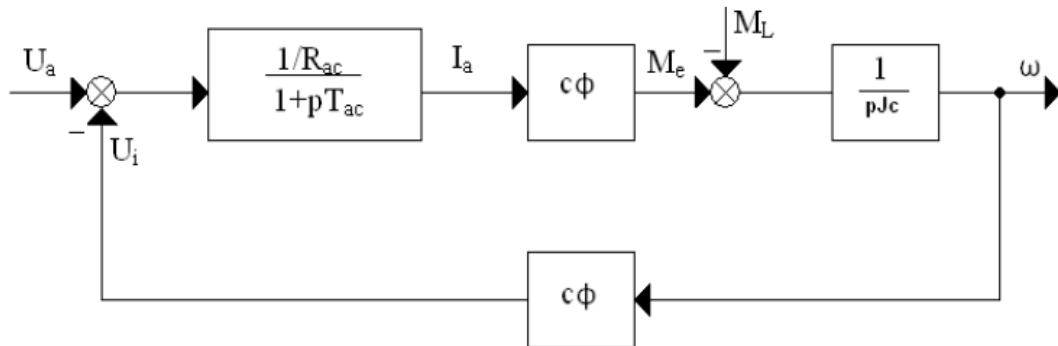
Rovnice po Laplaceovy transformaci:

$$U_a(p) = U_i(p) + R_{ac}I_a(p) + L_{ac}(p)I_a(p) \quad (5.6)$$

$$U_i(p) = c\Phi\omega(p) \quad (5.7)$$

$$M_e(p) = J_c p \omega(p) + M_L(p) \quad (5.8)$$

Z těchto rovnic složíme blokové schéma ss pohonu s cizím buzením, které bude využito v program Matlab-Simulink. [6]



Obr. 5.1 Blokové schéma stejnosměrného motoru s cizím buzením

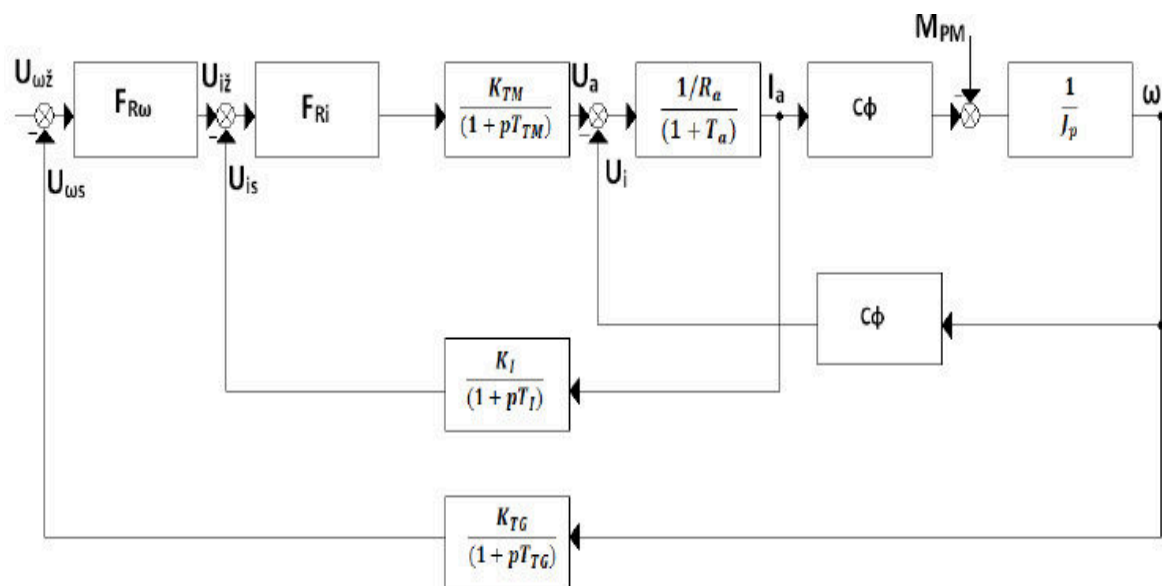
6. Aplikace metod na příkladu

V této kapitole bude popsána aplikace estimace rychlosti a aplikace vybraných neuronových regulátorů v řízení rychlosti stejnosměrného pohonu. Pro modelování a simulaci regulačních struktur je použit program Matlab-Simulink.

6.1 UNS jako estimátor rychlosti

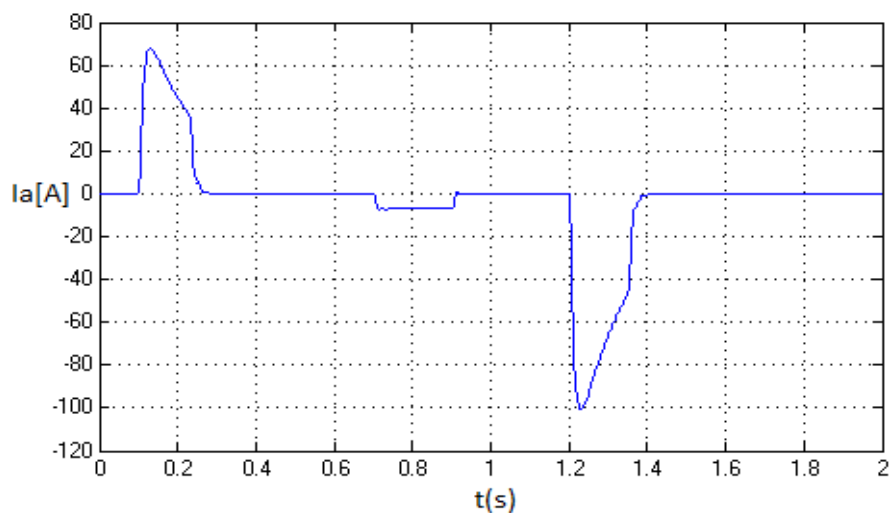
Jedna z aplikací umělé neuronové sítě pro řízení elektrického pohonu je jako estimátor rychlosti. Estimace znamená pozorování, to znamená, že snímáme různé veličiny stejnosměrného motoru, pomocí kterých pak naučíme umělou neuronovou síť vypočítávat otáčky. Při snímání otáček pomocí UNS, vypojíme snímač otáček (v tomto případě tachodynamo) zapojeného ve zpětné vazbě a nahradíme ji naučenou sítí. Jedná se tudíž o bezsenzorové řízení otáček.

Při aplikování metody budeme vycházet z blokového schématu pro řízení rychlosti stejnosměrného motoru.

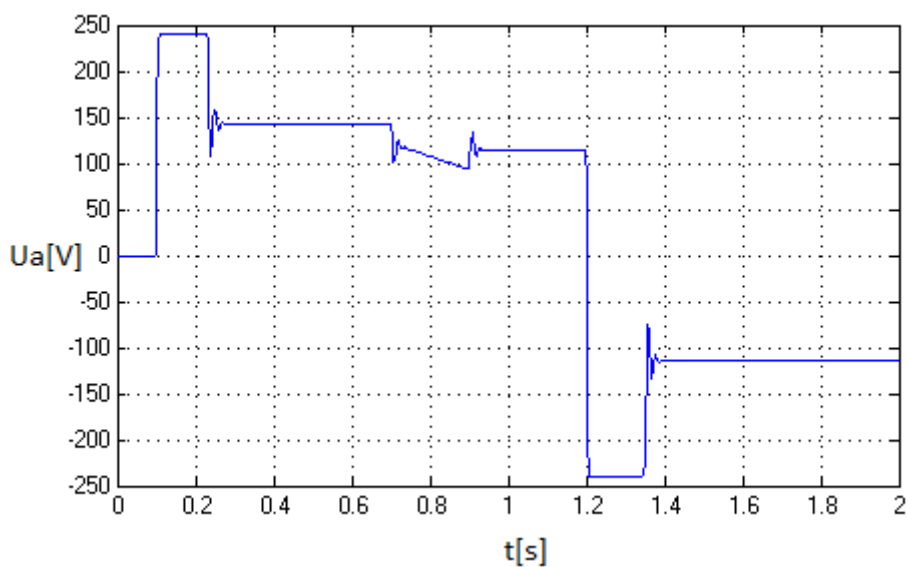


Obr. 6.1 Blokové schéma řízení rychlosti

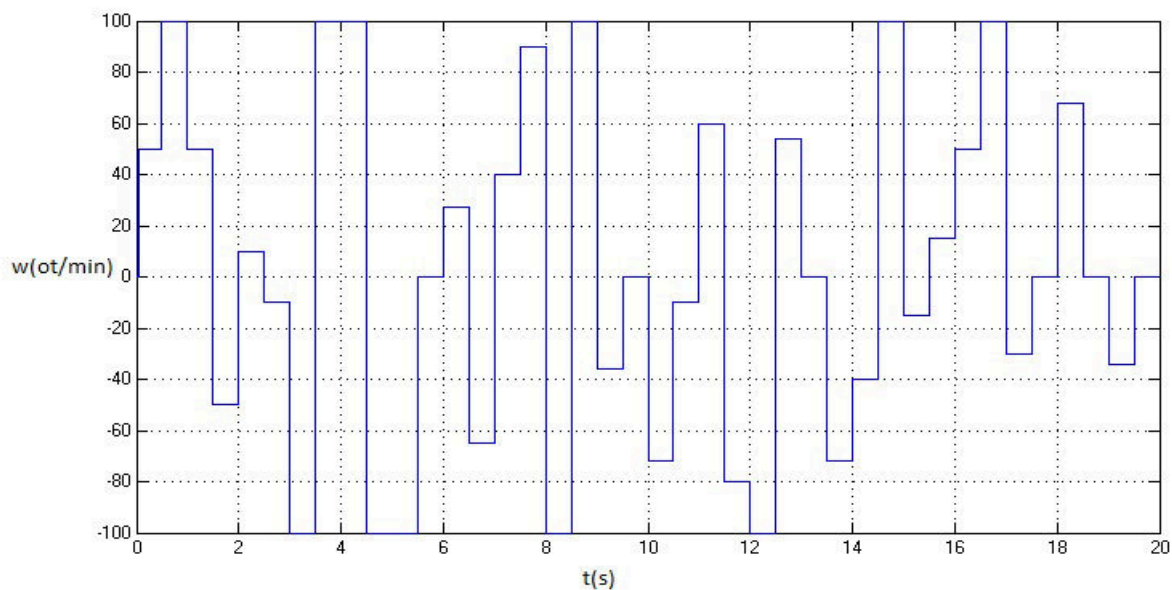
Pro estimaci rychlosti snímáme napětí kotvy U_a , proud kotvy I_a a jejich dvě předchozí hodnoty, nasnímané pomocí zpožďovacích bloků, a také výsledné otáčky za tachodynamem. Celkově bylo nasbíráno kolem 100 000 trénovacích dat.



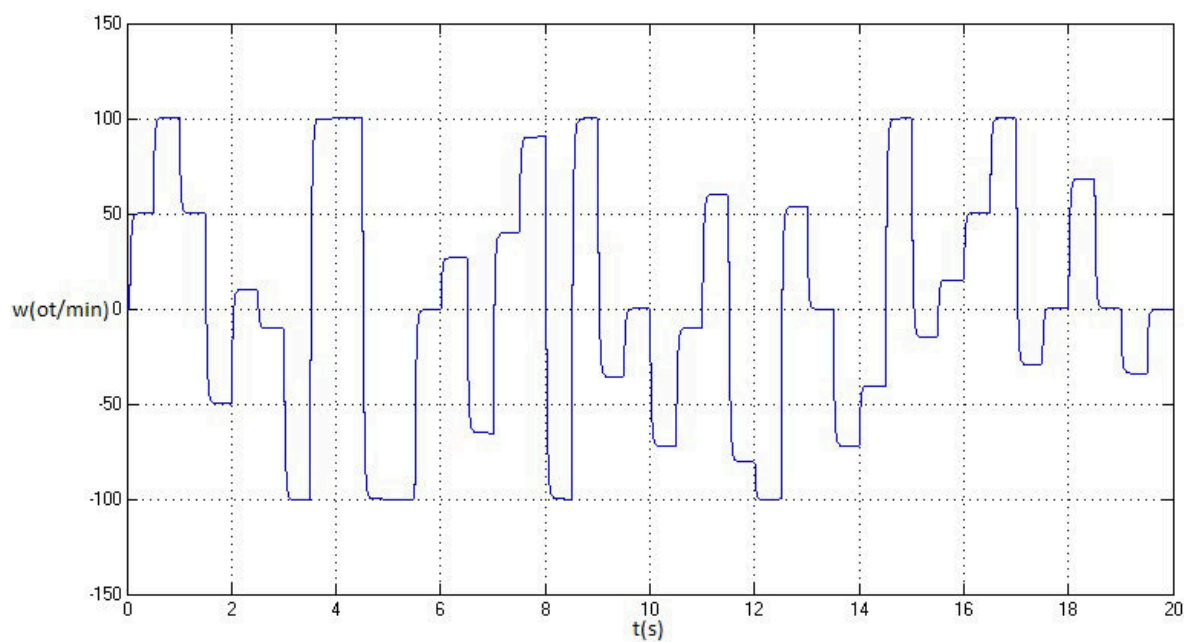
Obr. 6.2 Vzorový nasnímaný průběh proudu kotvy



Obr. 6.3 Vzorový nasnímaný průběh napětí kotvy



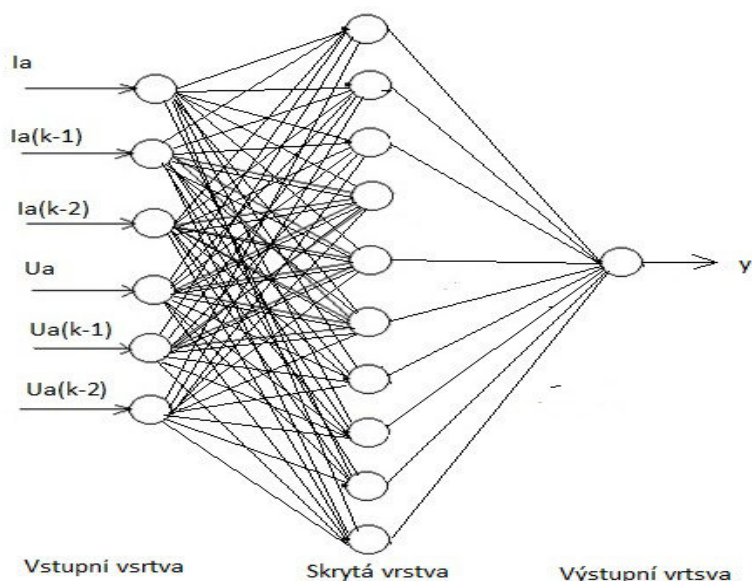
Obr. 6.4. Požadované otáčky a jejich skokové změny, které byly použity pro trénování UNS.



Obr.6.5. Skutečné otáčky použité jako výstupní trénovací data pro UNS.

Trenovací data byla pro další práci uložena ve Workspace v programu Matlab. Pro vytvoření neuronové sítě a její následné trénování byl použit NN toolbox obsažený v programovém prostředí MATLAB. Síť byla vytvořena jako dvouvrstvá dopředná síť s jednou skrytou vrstvou. Struktura

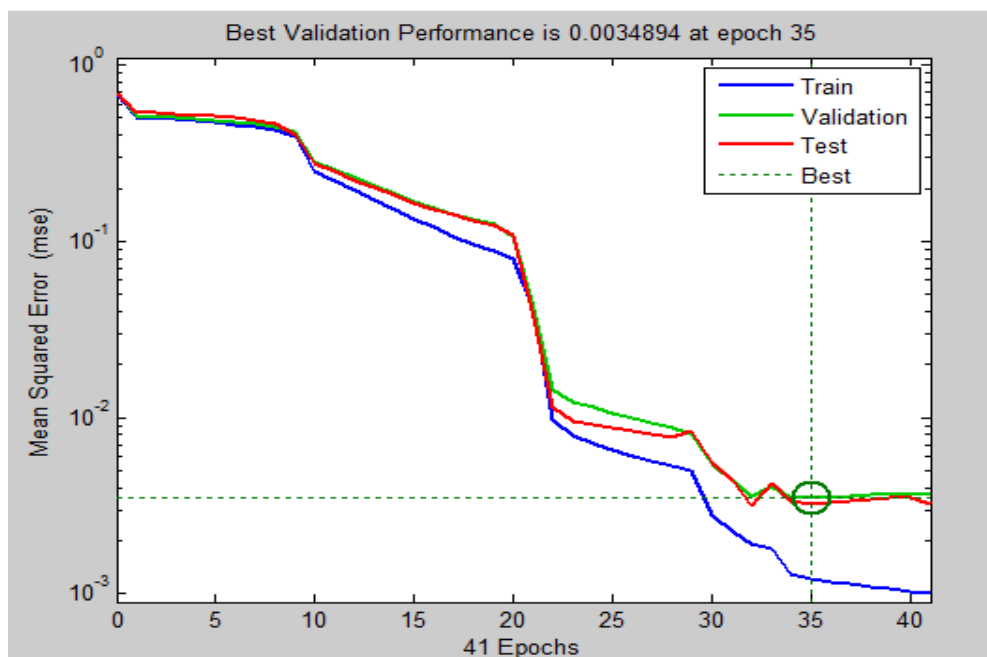
neuronové sítě je vyobrazena na obr. 6.6, pro ukázkou je zde neuronová síť s 10 neurony ve skryté vrstvě.



Obr. 6.6 Struktura použité neuronové sítě s 10 neurony ve skryté vrstvě

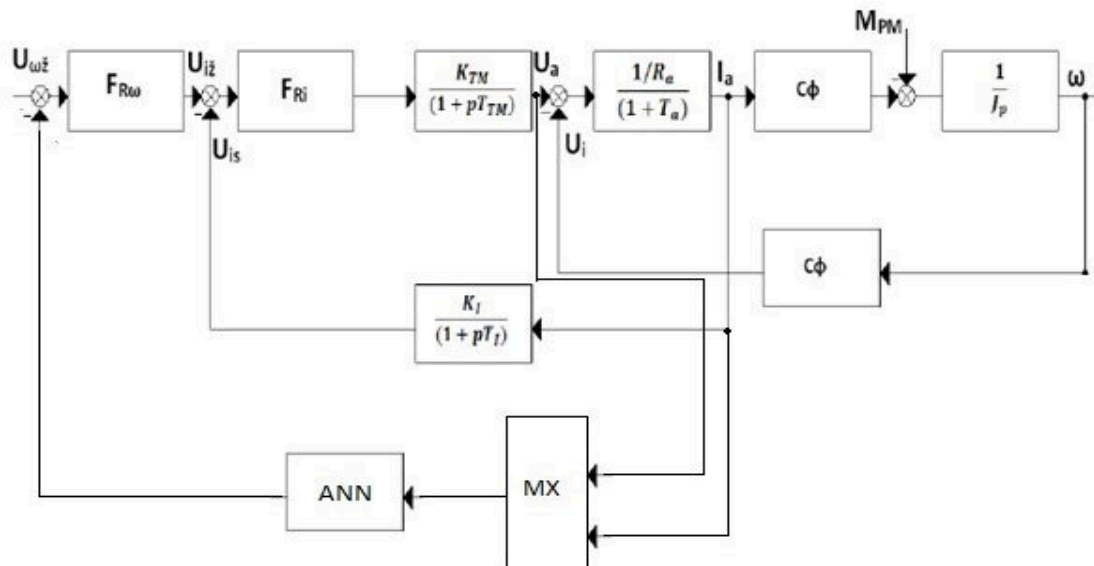
Neuronová síť byla vytvořena jakou dvouvrstvátá dopředná síť se sigmoidní aktivační funkcí pro neuron ve skryté vrstvě a lineární aktivační funkcí pro výstupní vrstvu. Síť byla trénovaná Levenberg-Marquardtovou metodou. A nastavena na 1000 trénovacích dat.

Na obr. 6.7. jsou zobrazeny výsledky trénování neuronové sítě s 10 neurony ve skryté vrstvě. Vidíme, že trénování bylo ukončeno po dosažení minimální chyby epochy, které bylo dosaženo u 35 epochy.

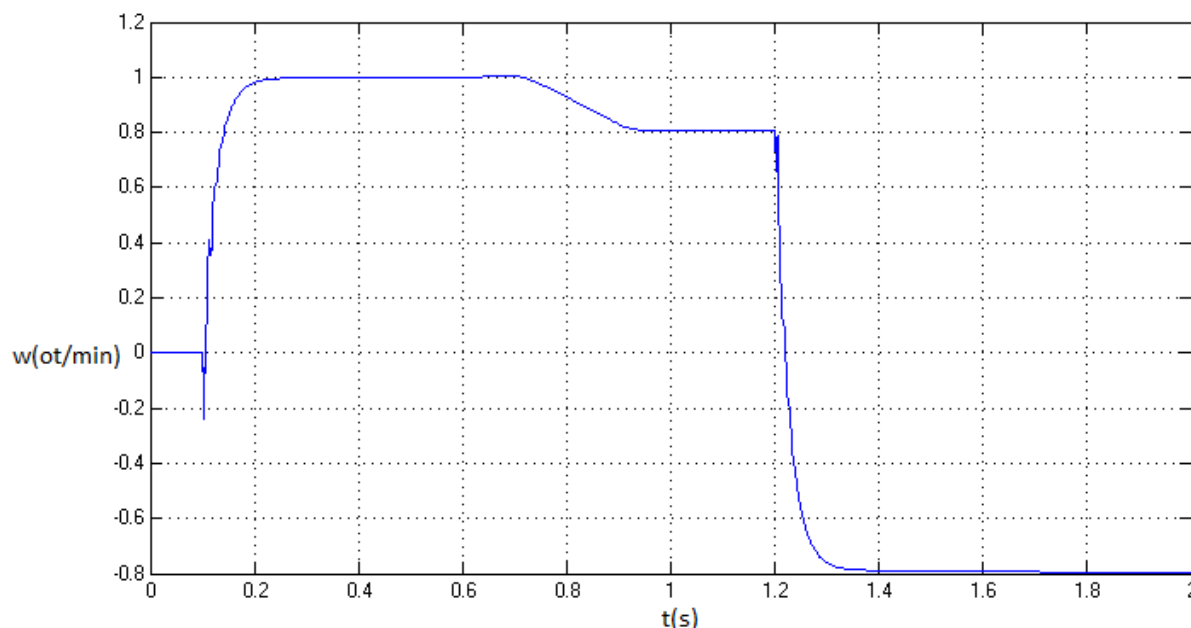


Obr. 6.7 Průběh trénování neuronové sítě s 10 neurony ve skryté vrstvě

Po natrénování sítě byla síť zapojena do regulační struktury podle schématu vyobrazeném na obr. 6.8. Kdy jak bylo na začátku kapitoly popsáno, je tachodynamo, které se běžně používá v regulační smyčce pro snímání otáček nahrazeno námi naučenou UNS, která pomocí vstupních dat (U_a , I_a) vypočítává otáčky motoru.



Obr. 6.8. Blokové schéma estimace rychlosti za pomoci UNS.



Obr. 6.9. Výstup z neuronové sítě

Na obr. 6. 9. je reakce naučené neuronové sítě na trénovací data. Signál (požadované otáčky), byly zvolené pro názornost jiné, aby bylo vidět, že naučená neuronová síť je schopná estimovat i otáčky na které nebyla naučena. Po zapojení neuronové sítě do zpětné vazby vykazuje regulační struktura překmity žádaných otáček. Tyto překmity je třeba odfiltrovat filtračním členem. Při použití filtračního členu, který byl zapojen za neuronovou síť, jsme vycházeli z jednoduché rovnice:

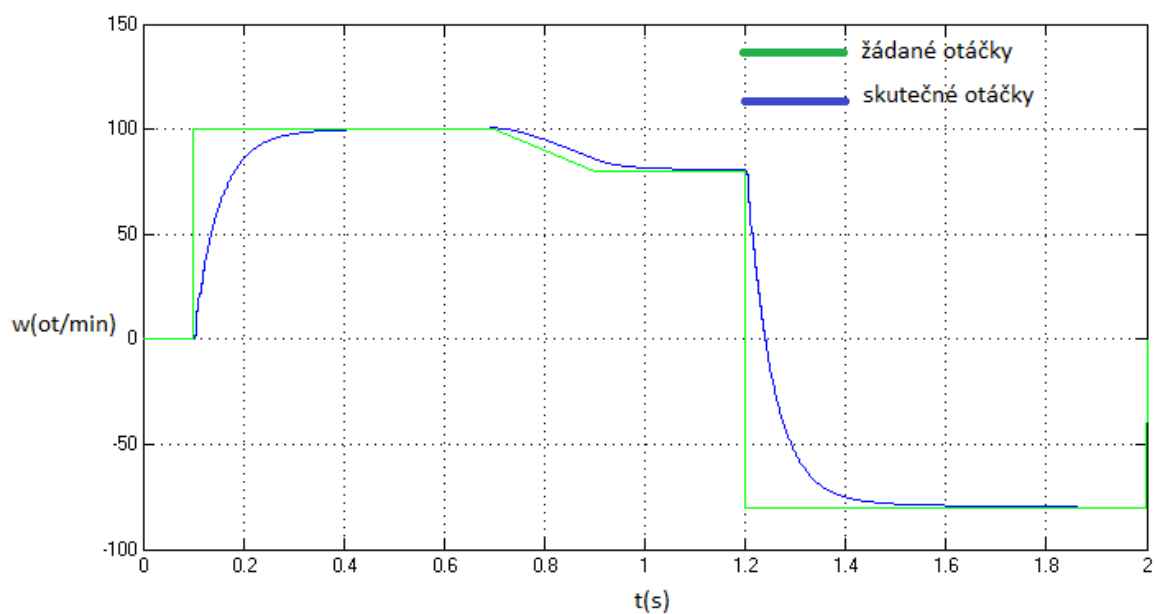
$$f = \frac{y(k)+y(k-1)}{2} \quad 6.1$$

kde f Filtr

$y(k)$ Hodnota z výstupu neuronové sítě

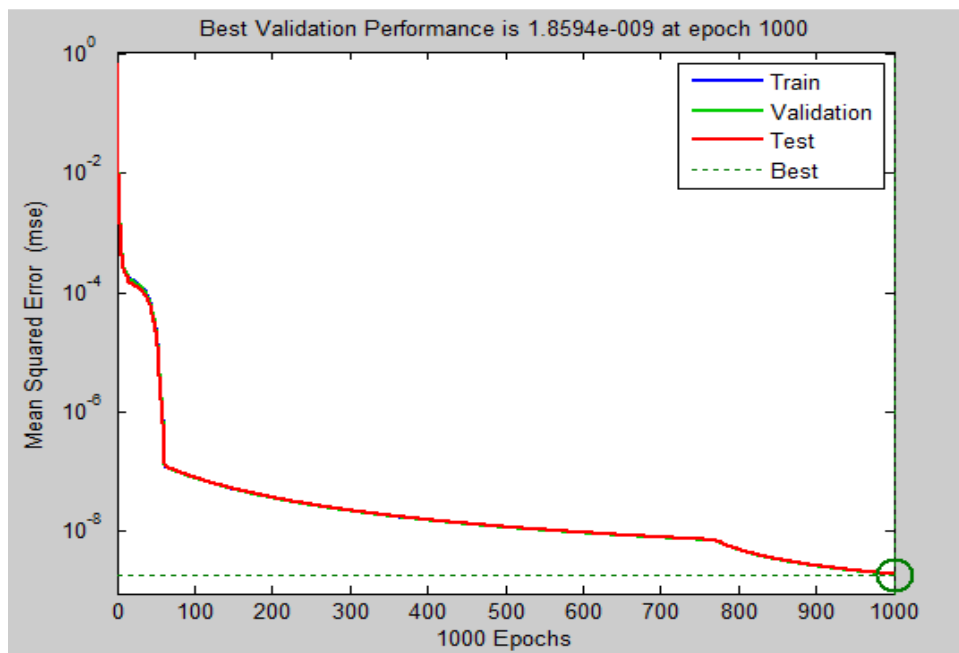
$y(k-1)$ Předchozí hodnota z neuronové sítě

Na následujících průbězích jsou zobrazeny průběhy, kdy do zpětné vazby v regulační smyčce byla zapojena neuronová síť, která otáčky estimovala již se zapojeným výše zmíněným filtrem. Oproti obr. 6.9, kde je průběh UNS bez filtru lze vidět na obr. 6.10, že zmíněný filtr zpomaluje dobu regulace, což je s teorie filtrace nežádoucích jevů při regulaci, očekávané.

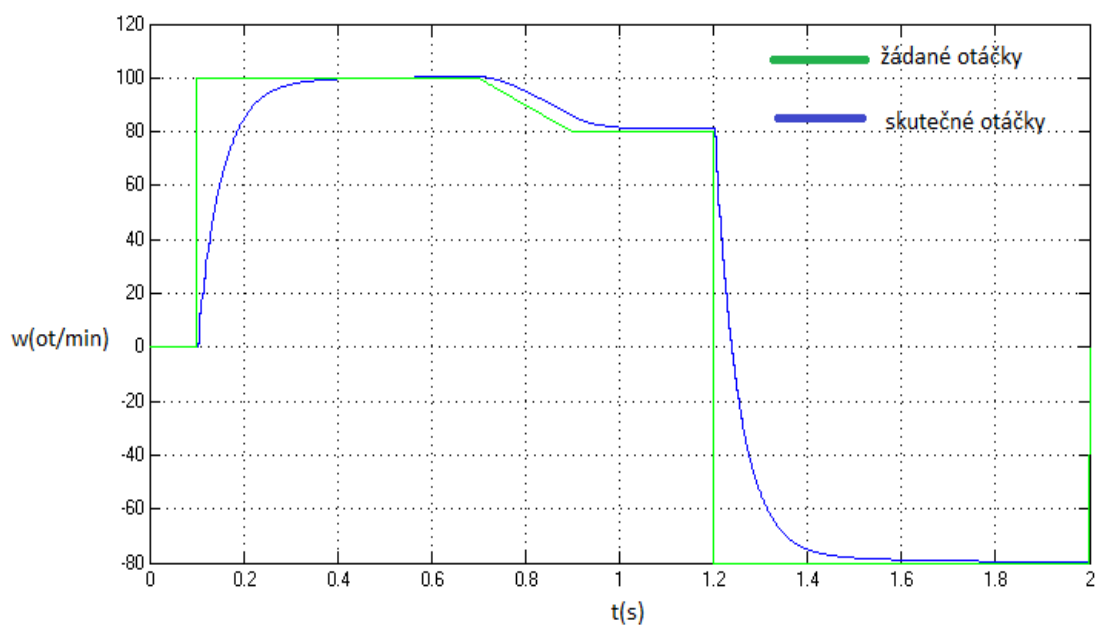


Obr. 6.10. Průběh otáček s estimátorem s 10 neurony ve skryté vrstvě

Dalším cílem pro estimaci bylo zjistit při kolika neuronech ve skryté vrstvě, byla síť již schopná estimovat otáčky. Proto byl hned vyzkoušen pouze jeden neuron ve skryté vrstvě. Síť byla vytvořena stejně, jako je struktura na obr.6.6., ale ve skryté vrstvě byl pouze jeden neuron. Z obr. 6.11. vidíme, že při použití jednoho neuronu ve skryté vrstvě bylo trénování ukončeno až při dosažení maximální počtu epoch, která byla nastavena na 1000 epoch.

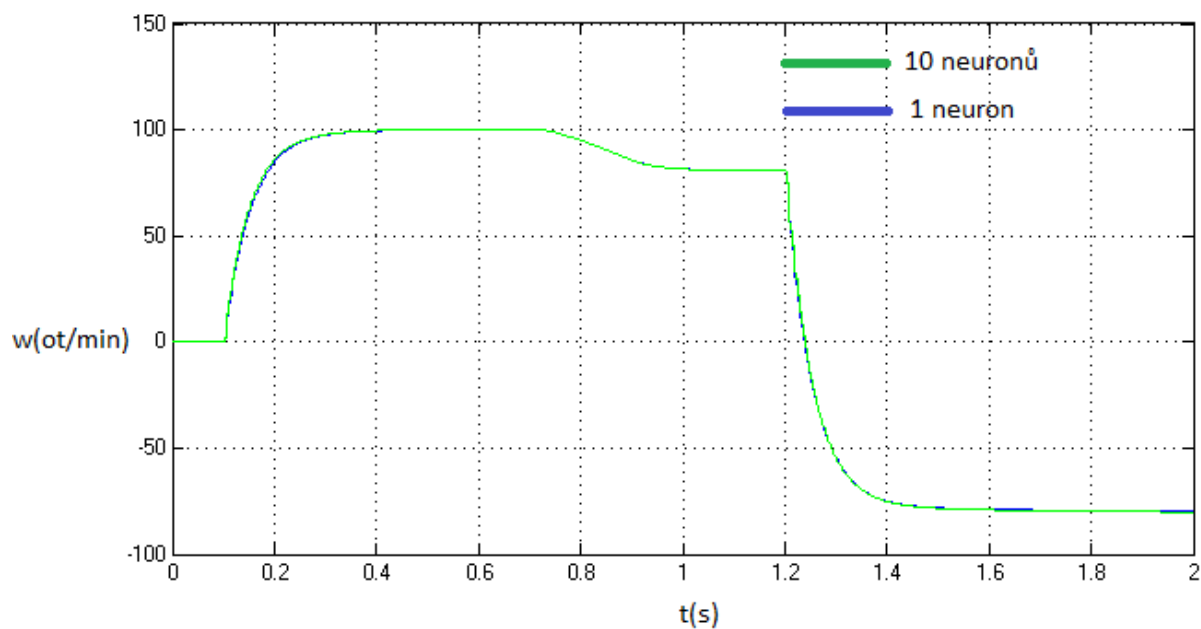


Obr. 6.11. Průběh trénování sítě pro 1 neuron ve skryté vrstvě

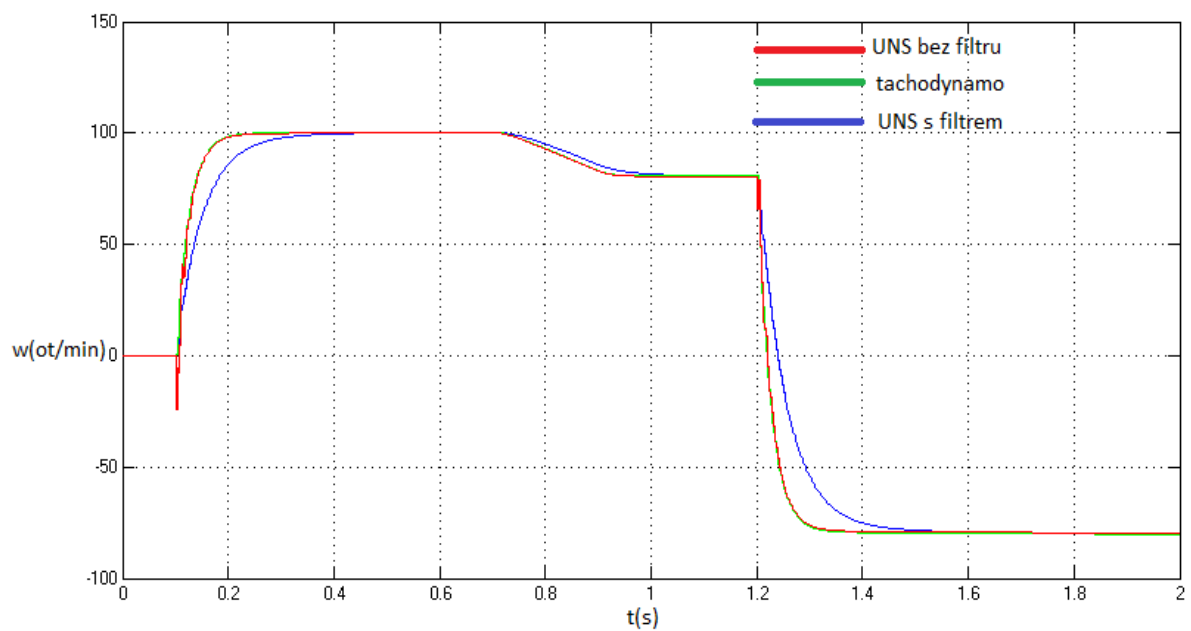


Obr. 6.12 průběh otáček s estimátorem s 1 neuronem ve skryté vrstvě

Z posledních dvou průběhů, lze vidět, že řízení otáček s 10 a s 1 neuronem ve skryté vrstvě se moc neliší, je zde pouze malá odchylka, kterou lze považovat za zanedbatelnou, či dokonce nulovou. Pro přesnější představu odchylky jsou tyto dva průběhy porovnány na obr. 6.13.



Obr. 6.13 porovnání průběhy otáček s různým počtem neuronů ve skryté vrstvě

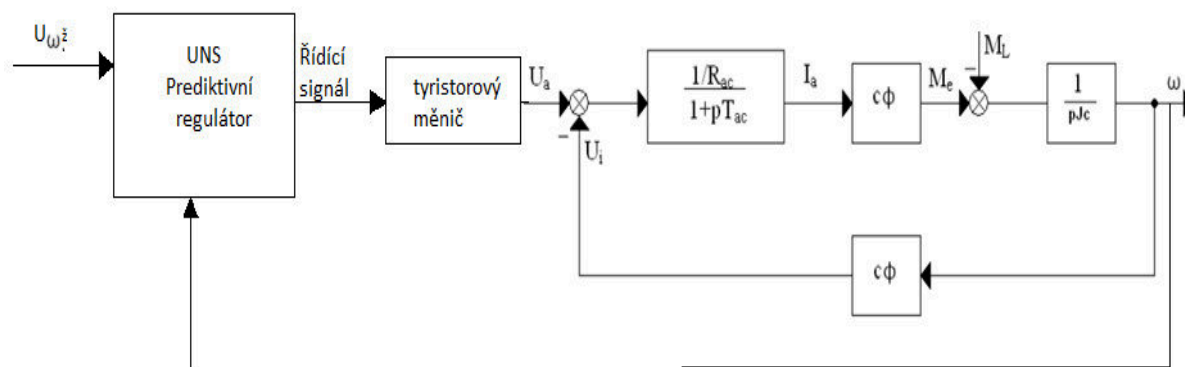


Obr. 6.14. Porovnání otáček s použitím tachodynáma, UNS bez filtru a UNS s filtrem

Na obr. 6.12 je zobrazeno porovnání průběhu otáček motoru za použití tachodynáma ve zpětné vazbě, UNS bez filtru a UNS s filtrem. Je zřejmé, že UNS je dobře naučena a estimuje otáčky tak jak byly nasnímany za použití tachodynáma ovšem jeví značný překmit. Ten byl odfiltrován filtrem zapojeným za UNS, který ovšem zpomaluje dobu regulace.

6.2 Prediktivní řízení za pomoci UNS

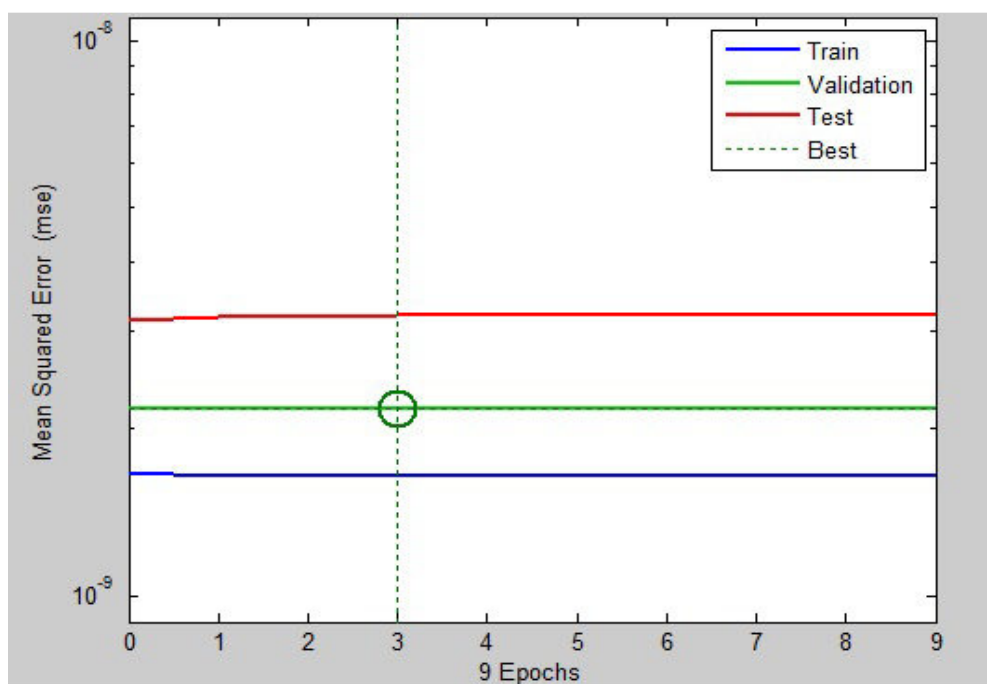
U prediktivního řízení pracujeme podle blokového schématu na obr.6.16. Celá tato regulační smyčka je realizována v program Matlab-Simulink. Predictive controller neboli prediktivní regulátor je obsažen v Neural Network toolbox. Neuronová síť je zapojena na místo regulátoru otáček, to znamená, že už se jedná o neuronový regulátor. Použitá síť je dopředná vícevrstvá síť se sigmoidní funkcí pro neurony ve skryté vrstvě a lineární funkcí pro výstupní neuron. Skrytá vrstva byla nastavena na 5 neuronů, vzorkovací perioda byla zvolena na 0.005s. Bylo nasbíráno 50 000 trénovacích vzorků a počet trénovacích epoch byl nastaven na 1000. To vše se nastavuje přímo v bloku controlleru.



Obr. 6.16 blokové schéma prediktivního řízení

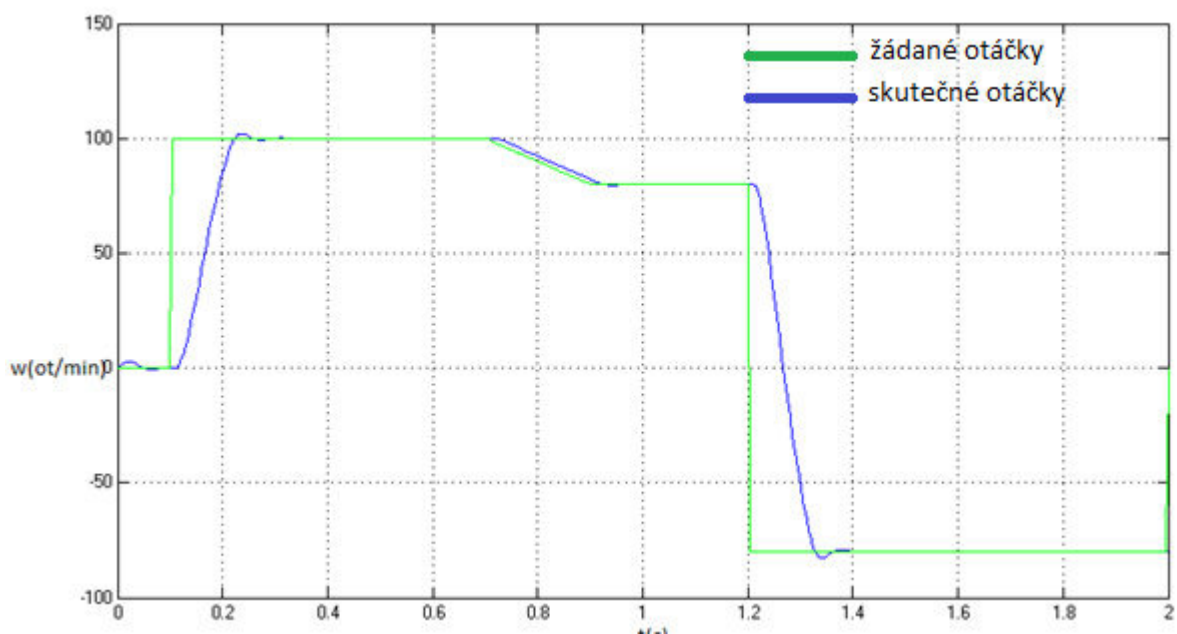
Do controlleru vstupuje referenční signál, což je námi zvolený zdroj a výstup z motoru v podobě otáček, výstupem je pak kontrolní signál, který řídí otáčky motoru. Popis principu prediktivního řízení je uveden v **kapitole 3.2.5**.

Stejně jako u estimátoru rychlosti, bylo zkoumáno při jakém počtu neuronů ve skryté vrstvě je regulátor schopný řídit otáčky motoru. Bylo vyzkoušeno 5,3 a 1 neuron ve skryté vrstvě.

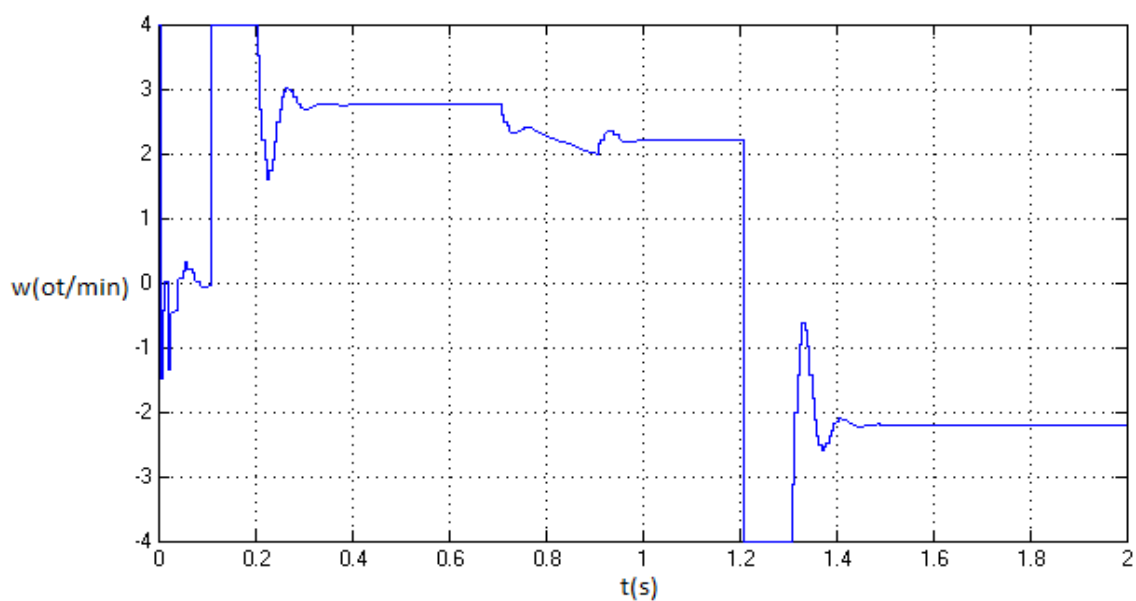


Obr. 6.17 průběh trénování sítě pro 5 neuronů ve skryté vrstvě

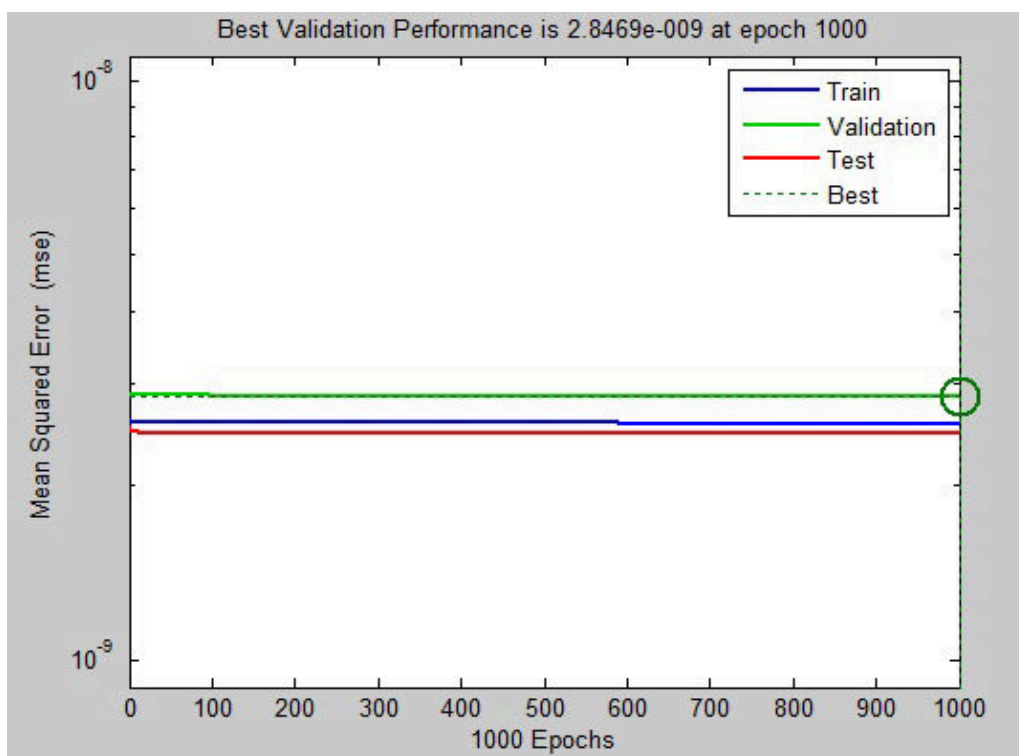
U obr. 6.17 vidíme, že trénování bylo ukončeno po dosažení minimální chyby epochy, která byla dosažena již u třetí epochy.



Obr. 6.18 Průběh prediktivního řízení s 5 neurony ve skryté vrstvě

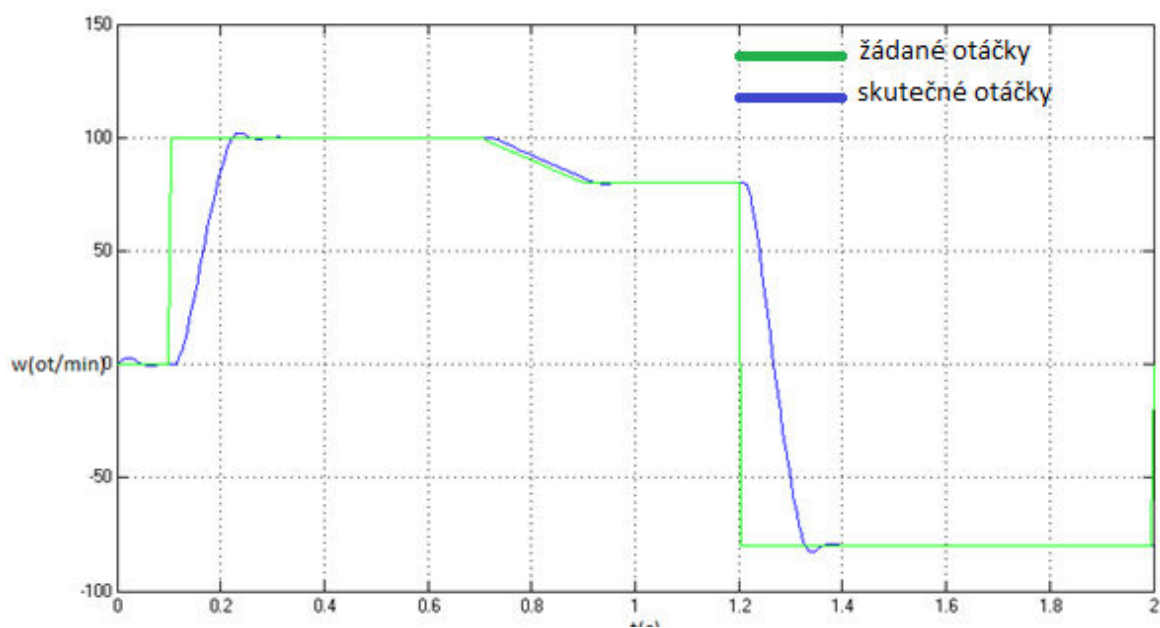


Obr.6.19 Control signál vstupující do motoru, při 5 neuronech ve skryté vrstvě

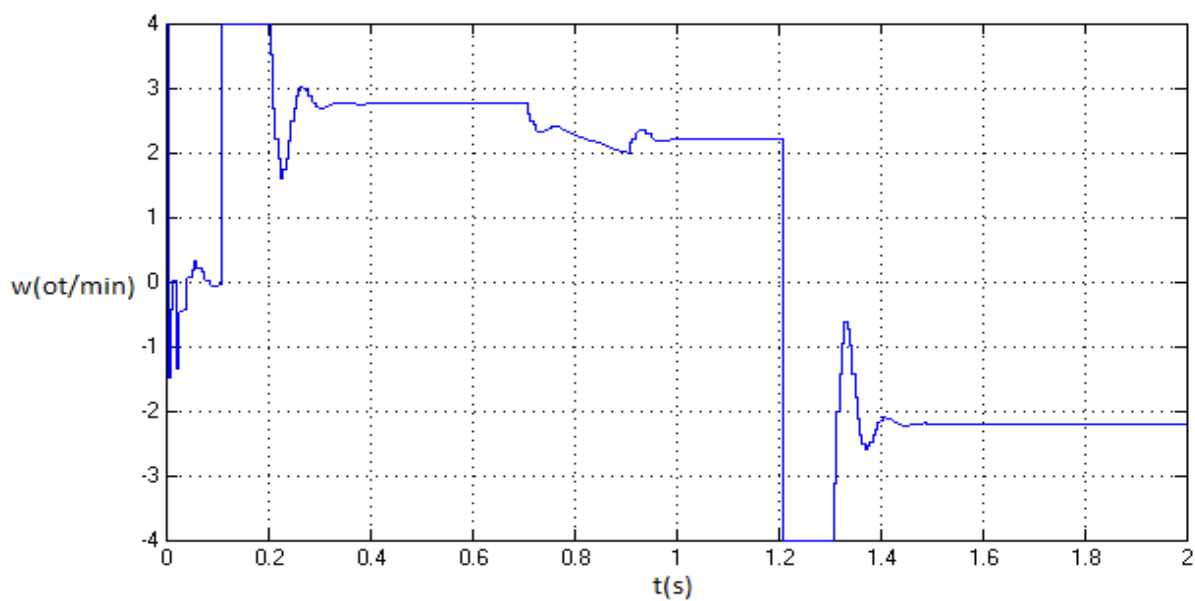


Obr. 6.20 průběh trénování sítě pro 3 neuronů ve skryté vrstvě

Z obr. 6.18 vidíme, že trénování sítě při 3 neuronech ve skryté vrstvě bylo ukončeno po dosažení maximálního počtu epoch, které bylo, jak už bylo zmíněno nastaveno na počet tisíc.

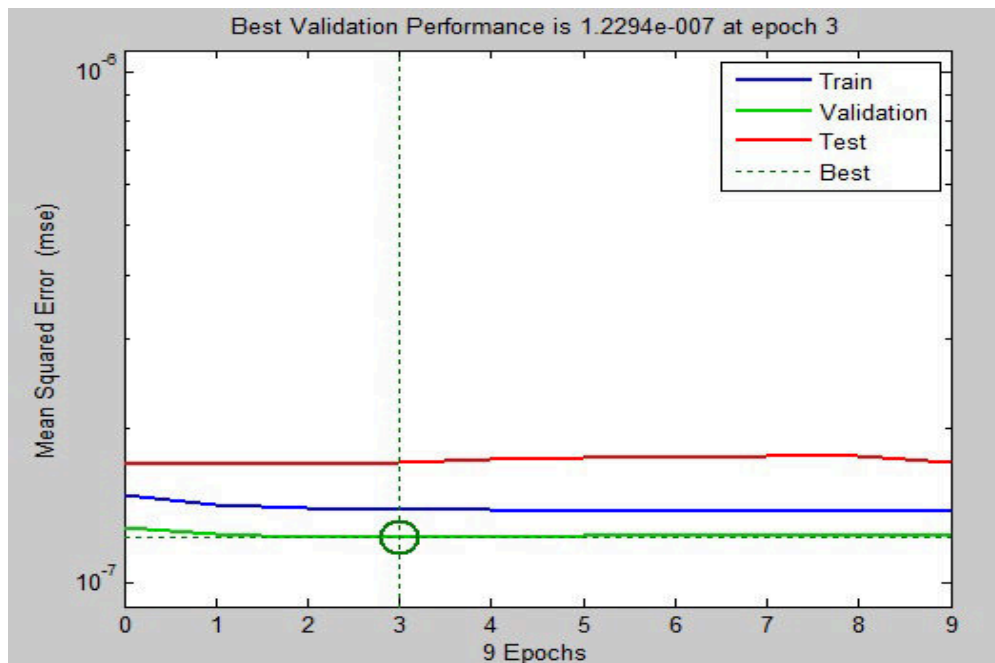


Obr. 6.21 Průběh prediktivního řízení se 3 neurony ve skryté vrstvě



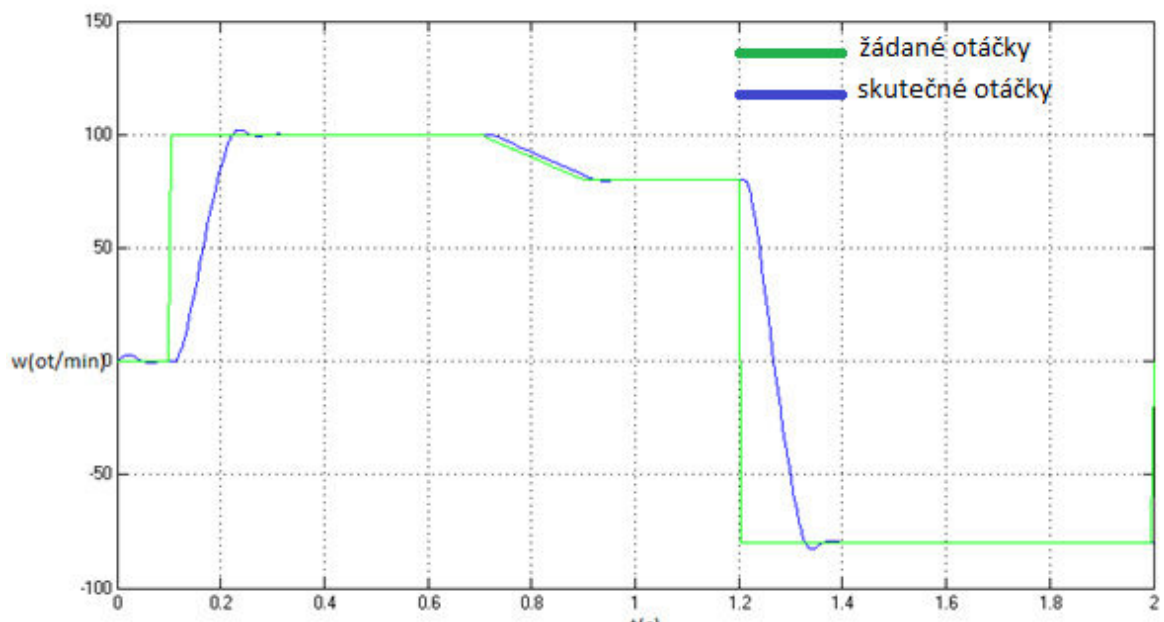
Obr.6.22. Control signál vstupující do motoru, při 3 neuronech ve skryté vrstvě

Z průběhů na obr. 6.18. a obr.6.21. vidíme, že se přechodová charakteristika vůbec neliší a proto je jednodušší zvolit variantu se 3 neurony ve skryté vrstvě. Nadále tedy bude vyzkoušena varianta s 1 neuronem ve skryté vrstvě.

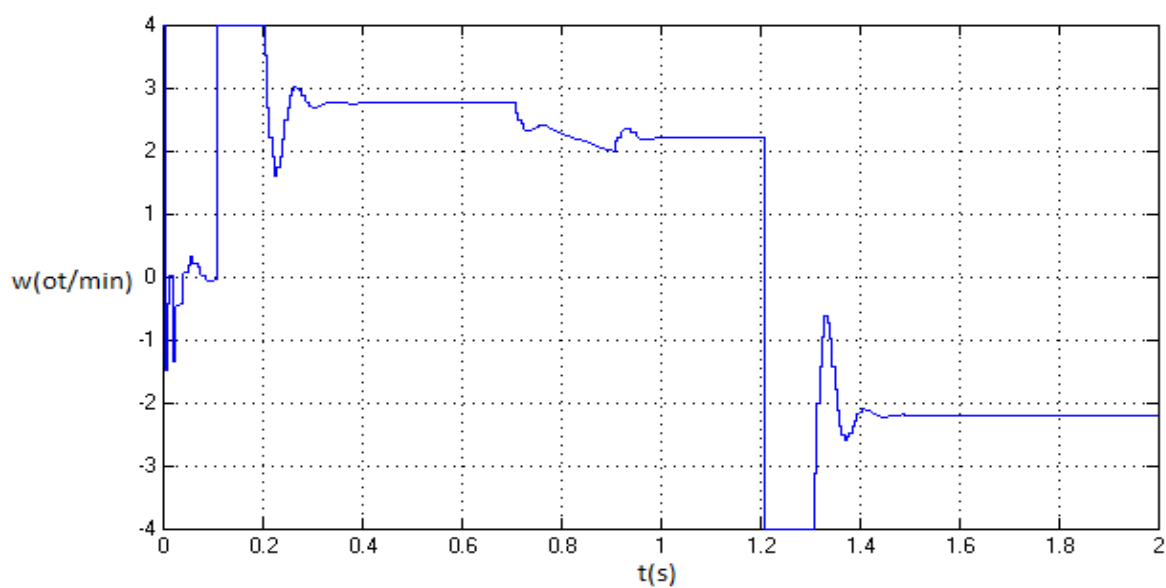


Obr. 6.23 průběh trénování sítě pro 1 neuron ve skryté vrstvě

Trénování sítě pro 1 neuron ve skryté vrstvě bylo ukončeno podosažení minimální chyby epoch, která byla stejně jako u trénování s 5 neurony při třetí epoše.



Obr. 6.24 Průběh prediktivního řízení s 1 neuronem ve skryté vrstvě



Obr.6.25 Control signál vstupující do motoru, při 1 neuronu ve skryté vrstvě

Porovnáme-li všechny přechodové charakteristika s obr.6.18. , obr.6.21 a obr.6.24, lze vidět, že průběhy jsou totožné. Znamená to tedy, že pro prediktivní řízení můžeme použít stejně taktéž jeden neuron ve skryté vrstvě. Použití různých počtů neuronu se liší pouze ve výsledku trénování sítě což jde vidět z obr.6.17. , obr.6.20. a obr.6.23.

6.3 Zhodnocení použitých UNS

Zhodnotíme-li UNS použitou jako estimátor rychlosti tak můžeme vidět z obr.6.9., že bez použití filtračního členu je regulovaný průběh rozkmitaný. Proto je zde nutno použít filtrační člen dle vzorce 6.1. Tento filtrační člen odstraní nežádoucí překmity, ale jak lze vidět na obr. 6.10 a 6.12, tak navíc zpomaluje dobu regulace, tj. dobu za kterou se otáčky ustálí na požadované hodnotě. Cílem bylo i zkoumat, při jakém počtu byla neuronová síť schopna otáčky estimovat. Prvně byla vyzkoušena struktura s 10 neurony ve skryté vrstvě, jejíž struktura je na obr. 6.6. Tento počet byl zvolen, jelikož se jevil jako dostačující a byla vyzkoušena síť i s více neurony, ale výsledek skutečných otáček byl již neměnný. Proto bylo vyzkoušeno při jakém minimálním počtu neuronů ve skryté vrstvě UNS bude otáčky estimovat, proto byla ihned vyzkoušena síť s 1 neuronem ve skryté vrstvě a ukázalo se, že i s jedním neuronem je síť schopna otáčky estimovat téměř stejně jako s 10 neurony. Regulační odchylka mezi těmito dvěma strukturami neuronových sítí byla téměř nulová.

Na obr. 6.15 je přechodová charakteristika se zatížením pracovního mechanismu a lze vidět, že se regulační smyčka dokáže vypořádat s tímto problémem, který se může jevit i jako porucha a ustálí se po čase na požadované hodnotě. Jelikož je zpravidla cílem při řešení neuronových sítí být schopen najít co nejjednodušší strukturu neuronové sítě, nic nám tedy podle simulačních výsledků nebrání v použití UNS jako estimátor rychlosti s 1 neuronem ve skryté vrstvě.

U prediktivního řízení rychlosti, bylo taktéž vyzkoušeno při jakém množství neuronů ve skryté vrstvě, je regulátor schopný otáčky estimovat. Vyzkoušeno bylo prediktivní řízení s 5,3 a 1 neuronem ve skryté vrstvě. Výsledné přechodové charakteristiky na obr.6.18. , obr.6.21. a obr.6.24. poukazují na to, že taktéž jako u použití UNS jako estimátoru rychlosti, nám vystačuje ta nejjednodušší varianta UNS a to s jedním neuronem ve skryté vrstvě. Odlišnost vykazuje pouze výsledek trénování neuronové sítě.

7. Závěr

Cílem této diplomové práce bylo seznámení aplikace umělých neuronových sítí v řízení elektrických pohonů. Seznámit se základními znalostmi neuronové sítě a její struktury. Poté aplikovat vybrané metody použití neuronové sítě v řízení stejnosměrného elektrického pohonu. Metod aplikace UNS je v této práci teoreticky popsáno více, ale praktická část je zaměřena na použití UNS, jako estimátor rychlosti a použití neuronové sítě jako regulátor za pomoci prediktivního řízení, kdy modelování a simulace struktur byly provedeny za pomoci programu Matlab-Simulink.

První část je tedy zaměřena na rozbor pojmu jako biologická neuronová síť, ze kterých UNS vycházejí. Po vyrozumění základních pojmů neuronové biologické sítě, je zde popis umělého neuronu, což je základní stavební prvek umělé neuronové sítě. Popsána byla i základní struktura umělých neuronových sítí, především síť typu feed-forward neural network, neboli vícevrstvé dopředné síť, které se v aplikaci elektrických pohonů používají nejčastěji. Tyto síť se učí podle různých typů učících algoritmů, přičemž nepoužívanější je metoda zpětného šíření chyby, neboli metoda Backpropagation a její modifikace jako Kwazi-Newtonova metoda a sní vycházející Levenberg-Marquardtův algoritmus, kterým byly učeny síť používané v této práci. Existuje samozřejmě více metod, ale popsány v této práci jsou tyto, jelikož nám pro práci při řízení elektrických pohonů vystačí.

Druhá část se zabývá právě použitím UNS v řízení elektrických pohonů. Teoreticky je zde rozebráno použití UNS pro modelování a identifikaci systému, diagnostiky poruch, identifikaci stavových veličin a řízení pohonů. U řízení pohonů, byly popsány čtyři základní skupiny neuronových regulátorů jako: přímé inverzní řízení, řízení s referenčním modelem, řízení s interním modelem a prediktivní řízení a použití neuronové sítě při bezsenzorovém řízení, kdy namísto čidla otáček je do zpětné vazby zapojeno UNS jako estimátor rychlosti.

Třetí a poslední část je věnována aplikaci UNS na námi vybraném příkladu. Jako příklad byl vybrán stejnosměrný pohon nacházející se v laboratoři katedry Elektroniky na VŠB- TU Ostrava. Metody, které byly vybrány pro tuto práci jsou použití UNS jako estimátor rychlosti a použití UNS jako regulátoru za pomoci prediktivního řízení. Bylo i zkoumáno při kolika neuronech ve skryté vrstvě byla regulační smyčka schopna regulovat a estimovat požadované otáčky.

U estimátoru rychlosti, byly jako vstupní data snímány hodnoty proudu a napětí motoru a jejich předchozí dvě hodnoty. Jako výstupní data byly nasnímané skutečné otáčky za pomoci tachodynamu. Je ovšem nutno podotknout, že regulátory rychlosti a proudu byly nastaveny experimentálně, kdy byly použity přímo bloky PID regulátory v programu Simulink. Použitá síť byla

vytvořena jako vícevrstvá dopředná síť se sigmoidní aktivační funkcí pro neurony ve skryté vrstvě a lineární aktivační funkcí pro neurony ve výstupní vrstvě, síť byla trénována Levenberg-Marquardtovým algoritmem. Bylo vyzkoušeno, při jakém množství neuronů ve skryté vrstvě byla síť schopna otáčky estimovat. Jako první byla vyzkoušena síť s 10 neurony a poté s 1 neuronem ve skryté vrstvě. Výsledkem z těchto porovnání je, že nám nic nebrání použít jednodušší strukturu neuronové sítě a to s 1 neuronem ve skryté vrstvě, jelikož odchylka je zde téměř nulová. Ovšem je nutné použít filtr zapojený pro odstranění nežádoucích překmitů za neuronovou síť.

U prediktivního řízení, kdy jako regulátor byl použit UNS prediktivní regulátor jsme zkoušeli také při jakém počtu začne regulátor řídit řádně otáčky. Použitá síť byla vytvořena jako u estimátoru, tzn. dopředná síť se sigmoidní aktivační funkcí pro neurony ve skryté vrstvě a lineární aktivační funkcí pro výstupní neuron. Počet neuronů, pro které byla síť trénována je 5,3 a 1 neuron ve skryté vrstvě. Při použití jednoho neuronu ve skryté vrstvě, vykazuje regulační struktura stejné výsledky jako u použití UNS s více neurony ve skryté vrstvě, proto nám pro tuto regulaci vyhovuje opět ta nejjednodušší verze UNS a to s jedním neuronem ve skryté vrstvě.

Závěrem bych tedy shrnul UNS jako estimátor rychlosti tak, že při použití filtru, pro odstranění rozkmitů je doba regulace sice pomalejší, ale musíme brát v potaz výhody využití neuronové sítě a to ty, že redukuje cenu hardwaru, můžeme ji nasadit v agresivním prostředí, je spolehlivá a oproti tachodynamu nevyžaduje údržbu. Pakliže hledáme nejjednodušší strukturu neuronové sítě, není problém využít UNS jako estimátor rychlosti pouze s jedním neuronem ve skryté vrstvě.

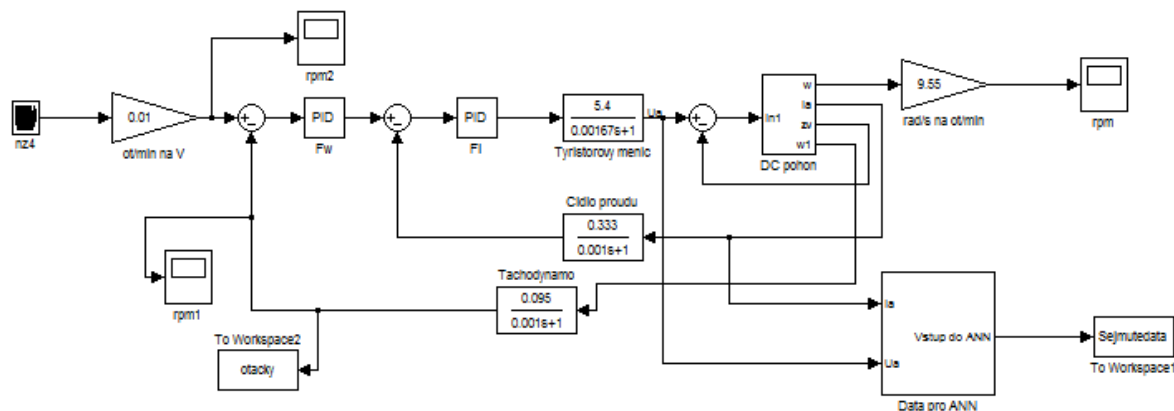
Prediktivní řízení založené na umělé neuronové síti vykazuje dobré regulační parametry, překmity jsou minimální a doba regulace je taky velmi dobrá. A jelikož i zde hledáme nejjednodušší strukturu neuronové sítě, můžeme tedy zůstat u struktury s 1 neuronem ve skryté vrstvě. Nevýhodou pak je jen to, že prediktivní řízení vyžaduje značné množství online výpočetního času, protože v každé vzorkovací periodě je vykonáván optimalizační algoritmus.

SEZNAM POUŽITÉ LITERATURY

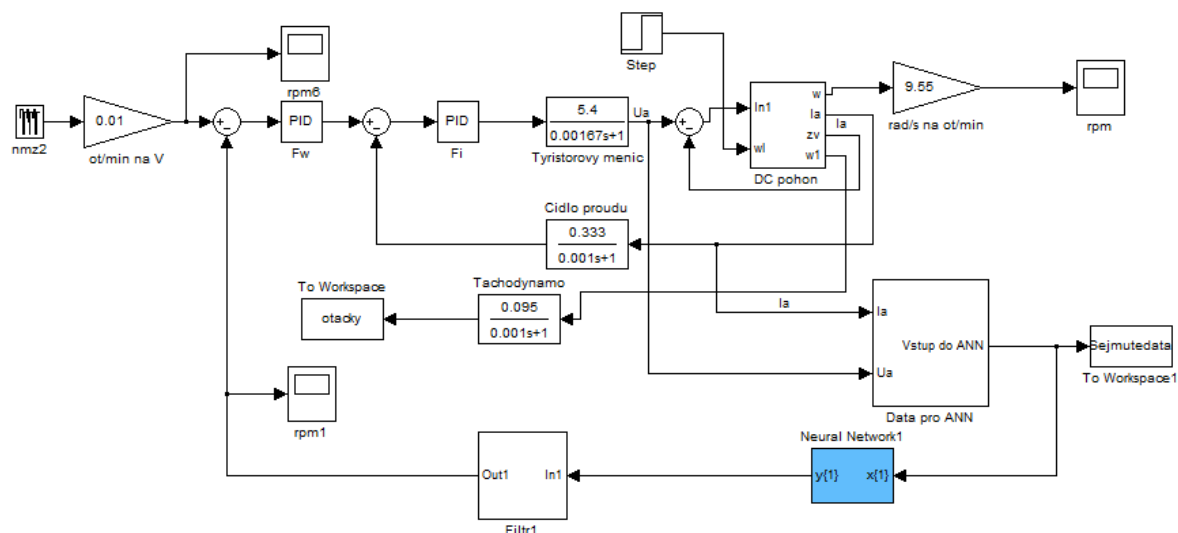
- [1] TIMKO, Jaroslav – ŽILKOVÁ, Jaroslava – BALARA, Dušan. *Aplikacie umelých neuronových sietí v elektrických pohonech*, Calypso s.r.o.,239s. ISBN 80-85723-27-1
- [2] NOVÁK, Mirko. *Umělé neuronové sítě: teorie a aplikace*, Praha,1998, 382s,ISBN :80-7179-132-6.
- [3] SCHMIDT, Michal. *Řízení procesů neuronovými sítěmi*. Brno 2004,66s. Diplomová práce na Fakultě informatiky a komunikačních technologiích na Vysokém učení technickém v Brně, Ústav automatizace a měřicí techniky
- [4] NEBORÁK,Ivo. *MODELOVÁNÍ A SIMULACE ELEKTRICKÝCH REGULOVANÝCH POHON*. VŠB-TU Ostrava 2002,172s.,ISBN: 80-248-0083-7
- [5]ČAJKA, Radim. *Aplikace umělé inteligence v elektrických regulovaných pohonech*. Ostrava 2007,113s Disertační práce na Fakultě elektrotechniky a informatiky na VŠB- Technická univerzita Ostrava.
- [6]ŠČAMBA,Jakub. *Aplikace různých typů regulátorů v řízení elektrických pohon*. Ostrava 2010. 46s. Bakalářská práce na Fakultě elektrotechniky a informatiky na VŠB- Technická univerzita Ostrava.
- [7] KUCHAR, M. *Řídicí aplikace neuronových sítí v elektrických regulačních pohonech*. Ostrava 2003, 101s.Disertační práce na VŠB- Technická univerzita Ostrava

PŘÍLOHY

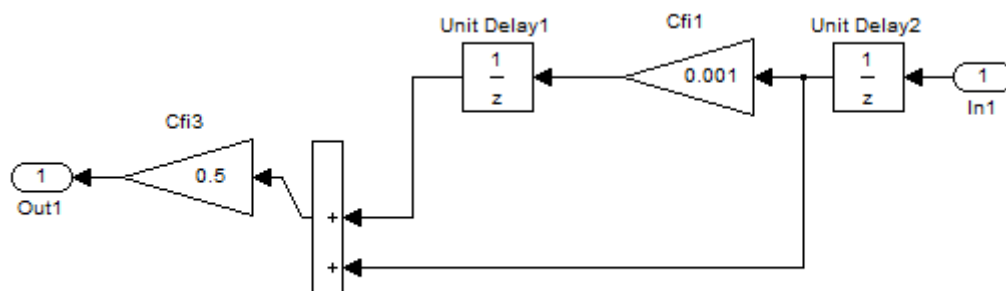
V příloze jsou zobrazeny simulační schémata pro sběr trénovacích dat pro UNS jako estimátor rychlosti (P.1) . Simulační schéma, kdy je ve struktuře řízení rychlosti zapojena již UNS (P.2). Náhled do simulačního schématu filtru použitého při simulaci UNS jako estimátoru rychlosti (P.3), schéma prediktivního řízení (P.4) a jako poslední schéma motoru použité v obou druhích simulace (P.5).



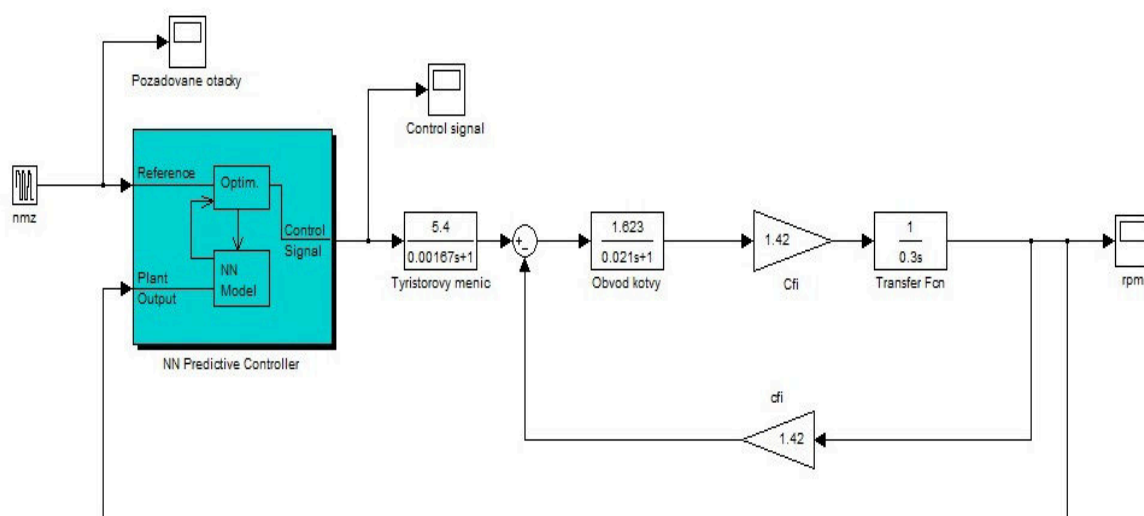
Obr.P.1 Simulační schéma pro sběr trénovacích dat



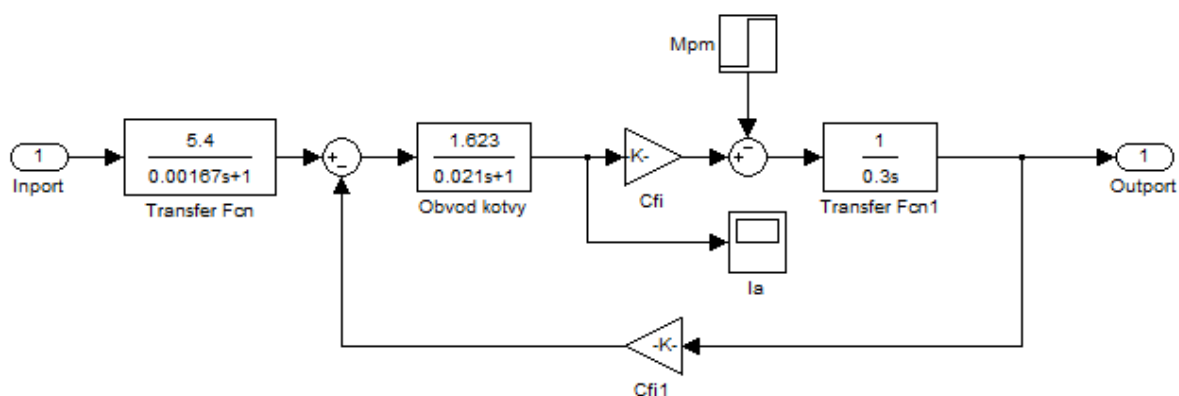
Obr.P.2 Simulační schéma UNS jako estimátoru rychlosti



Obr.P.3 Simulační schéma filtračního členu



Obr.P.4 Simulační schéma prediktivního řízení



Obr.P.5 Simulační schéma použitého stejnosměrného pohonu.